TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

SHRIRAM NANDAKUMAR

RECURRENT NEURAL NETWORKS FOR MULTI-MICROPHONE SPEECH SEPARATION

Master of Science thesis

Examiners: Prof. Tuomas Virtanen,
Dr.Pasi Pertilä
Examiners and topic approved by the
Faculty Council of the Faculty of
Computing and Electrical Engineering
on 5th October 2016

ABSTRACT

NANDAKUMAR, SHRIRAM: Recurrent neural networks for multi-microphone speech separation
Tampere University of Technology
Master of Science Thesis, 70 pages, 9 appendix pages
December 2017
Master's Degree Programme in Information Technology
Major: Signal Processing; Minor: Learning and Intelligent Systems
Examiners: Prof. Tuomas Virtanen, Dr. Pasi Pertilä
Keywords: Multi-microphone, microphone array, recurrent neural networks, long short-term memory, speech separation, CHiME-3, noise-robust speech recognition

This thesis takes the classical signal processing problem of separating the speech of a target speaker from a real-world audio recording containing noise, background interference — from competing speech or other non-speech sources —, and reverberation, and seeks data-driven solutions based on supervised learning methods, particularly recurrent neural networks (RNNs). Such speech separation methods can inject robustness in automatic speech recognition (ASR) systems and have been an active area of research for the past two decades. We particularly focus on applications where multi-channel recordings are available.

Stand-alone beamformers cannot simultaneously suppress diffuse-noise and protect the desired signal from any distortions. Post-filters complement the beamformers in obtaining the minimum mean squared error (MMSE) estimate of the desired signal. Time-frequency (TF) masking — a method having roots in computational auditory scene analysis (CASA) — is a suitable candidate for post-filtering, but the challenge lies in estimating the TF masks. The use of RNNs — in particular the bi-directional long short-term memory (BLSTM) architecture — as a post-filter estimating TF masks for a delay-and-sum beamformer (DSB) — using magnitude spectral and phase-based features — is proposed.

The data — recorded in 4 challenging realistic environments — from the CHiME-3 [1] challenge is used. Two different TF masks — Wiener filter and log-ratio — are identified as suitable targets for learning. The separated speech is evaluated based on objective speech intelligibility measures: short-term objective intelligibility (STOI) and frequency-weighted segmental SNR (fwSNR). The word error rates (WERs) as reported by the previous state-of-the-art ASR back-end — when fed with the test data of the CHiME-3 challenge — are interpreted against the objective scores for understanding the relationships of the latter with the former. Overall, a consistent improvement in the objective scores brought in by the RNNs is observed compared to that of feed-forward neural networks and a baseline MVDR beamformer.

PREFACE

This thesis is a spin-off from the TUT's submission to the CHiME-3 speech separation and recognition challenge. I am greatly indebted to Prof. Tuomas Virtanen for providing me an opportunity to work in the renowned Audio Research Group. Thanks to him for making me a part of the CHiME-3 challenge; for the regular meetings until settling down comfortably; and for providing me time, space and freedom to work on the topic.

My heartfelt gratitudes to Dr. Pasi Pertilä for lending a huge support throughout; for helping me in getting accustomed to Linux and cluster computing environment; for helping in the installation of software tools and sail through the hiccups thereof; for providing the necessary code base; for constantly making my learning curve steeper by coming up with challenging baselines to improve upon. Teaching has always been my first love — I again thank him immensely for providing me an opportunity to be a teaching assistant for the Audio and Speech processing course twice.

Acknowledgements to CSC — IT Center for Science, Finland, for the computational resources. Thanks to Gaurav for some fruitful discussions on similar topics. Thanks to Dr. Antti Hurmalainen for his collaboration during the CHiME-3 challenge participation. Thanks to the spirited members of the Audio Research Group — the half-yearly sauna events made unforgettable memories.

A big thanks to my dearest friends at this home away from home. Extended thanks to Amrita University, India and my colleagues there, especially to Prof. Sundararaman Gopalan.

Finally, I owe my most special thanks to my parents, my teachers, my sister, all my uncles and aunts and all my cousins. Special dedication to the dearest Krithika akka and Krupa — this thesis breathes their memories.

Shriram Nandakumar
Tampere, 6.12.2017

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Articulation index
AMS	Amplitude modulation spectrogram
ANSI	American National Standards Institute
ASA	Auditory scene analysis
ASR	Automatic speech recognition
BEM	Basic ensemble method
BLSTM	Bi-directional long short-term memory
BPTT	Backpropagation through time
BRNN	Bidirectional recurrent neural network
CASA	Computational auditory scene analysis
CMS	Cepstral mean subtraction
CMVN	Cepstral mean and variance normalization
CNN	Convolutional neural network
COLA	Constant overlap-add
DCT	Discrete cosine transform
DFT	Discrete Fourier Transform
DNN	Deep neural network
DoA	Direction of arrival
DSB	Delay-and-sum beamforming
FFNN	Feed-forward neural network
fMLLR	Feature-space maximum likelihood linear regression
fwSNR	Frequency-weighted segmental SNR
GCC	Generalized cross-correlation
GEM	General ensemble method
GFCC	Gammatone frequency cepstral coefficients
GMM	Gaussian mixture model
GRU	Gated recurrent unit
HMM	Hidden Markov model
IBM	Ideal binary mask
ILD	Inter-channel level difference
IPD	Inter-channel phase difference
IRM	Ideal ratio mask
LCMV	Linearly constrained minimum variance
LPC	Linear prediction coefficient
LSTM	Long short-term memory

MLP	Multi-layer perceptron
MFCC	Mel frequency cepstral coefficient
MLLT	Maximum likelihood linear transformation
MMSE	Minimum mean squared error
MSE	Mean squared error
MVDR	Minimum variance distortion-less response
NMF	Non-negative matrix factorization
PCM	Pulse coded modulation
PHAT	Phase transform
PLP	Perceptual linear prediction
RASTA	Relative spectral transformation
RBM	Restricted Boltzmann machine
ReLU	Rectified linear unit
RNN	Recurrent neural network
SDR	Signal to distortion ratio
SII	Speech intelligibility index
SNR	Signal to noise ratio
SRP	Steered response power
STFT	Short time Fourier transform
STI	Speech transmission index
STOI	Short-term objective intelligibility
TDoA	Time difference of arrival
TDNN	Time-delay neural network
TF	Time-frequency
WER	Word error rate
WSJ	Wall street journal

1. INTRODUCTION

Humans have long fantasized machines that can understand speech, a natural form of communication innate to humans but not to machines. Automatic speech recognition (ASR) systems strive to achieve this by converting captured speech waveforms into their linguistic content. Despite its long history dating back to 1952, the field was in the middle of its long trajectory of development [2, 3] until only recently — with the advent of deep learning there has been a giant leap in the progress. The difficulty has been long attributed to the inherent variability of acoustic signals due to various sources. On one extreme, there are challenges brought in by speaker pace and accent; on the other, highly reverberant environments and the presence of multiple speech sources such as crowd noise in the background pose a different challenge altogether. Even the best systems tend to perform poorly when exposed to conditions they are not well tuned for.

Consider the cafeteria scenario in Figure 1.1. The device that seeks to recognize the speech of the target speaker is confronted with the *cocktail party problem* — of replicating the capability of human auditory system by focusing on one particular sound source, here the speaker. The observed acoustic waveform, apart from speaker idiosyncrasies, is altered by roughly three factors: *competing sources*, the *environment* and the *transmission channel* [4]. Competing sources can be natural and artificial such as moving vehicle, wind, loud-speaker, dishes and voices from competing speakers. Environment — here the cafeteria — is a physical location which introduces other acoustic phenomena such as echo, reverberation and attenuation. Finally, transmission channel encompasses electro-mechanical properties such as microphones' response, signal bandwidth, quantization, compression and other types of transmission errors. Isolating the desired speech signal from competing sources is of primary concern — this forms the crux of speech "separation" systems which act as a front-end to ASR engines while also finding major applications in hearing aids, communication systems and the likes.

The gold standard for assessing machine speech recognition has always been comparing it to human-level performance and machines have already been able to nearly achieve it in *clean speech* recognition [5]. Active efforts are in progress to

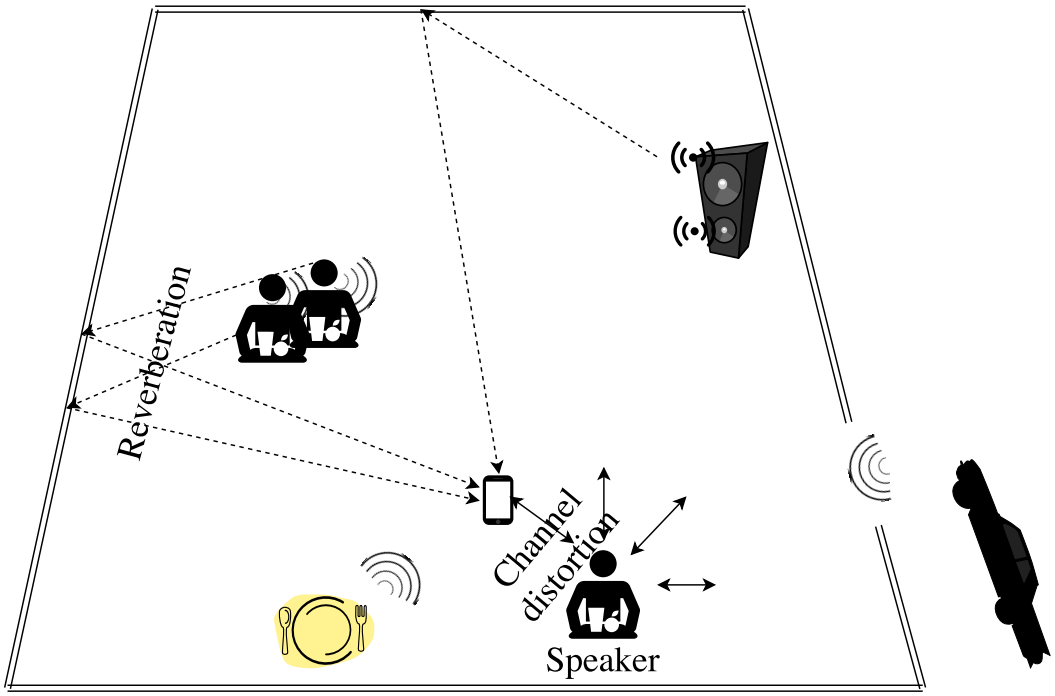


Figure 1.1 *A typical everyday noisy environment where speech separation is relevant*

achieve commensurate human parity in challenging real environments.

While building ASR systems using speech signals acquired by a single microphone is the most challenging, commercial products such as smart phones, tablet phones and laptops come with multiple microphones — often called *microphone arrays* — and they aid in achieving robustness by using directional cues and beamforming. The convergence of rather disparate array processing and ASR communities in developing noise robust ASR has bought in the possibility of using machine learning tools to jointly solve the problems. Along these lines, speech separation can be posed as a supervised learning problem and such a separation framework can be used as a front-end to ASR systems. In this thesis, speech separation is achieved by learning to predict an intermediate entity — called a TF mask or TF filter — which when applied to the mixture signal yields an estimate of the desired source signal.

The work for this thesis commenced from the participation of TUT's Audio Research Group in CHiME-3 speech separation and recognition challenge [1] where we suggested a sophisticated — yet slightly ad-hoc — framework for multi-microphone speech enhancement as applied to noise robust speech recognition (see Appendix). As a novice, my technical contributions to the submission involved only running batch scripts and scoring classifiers used in the pipeline. Post the submission, we

desired to "systematically" study the whole framework eventually leading to this thesis.

The primary goal of this thesis is thus to systematically study the following for CHiME-3 data:

- Time-Frequency masks as post-filters for a simple beamformer such as delay-and-sum beamformer (DSB).
- TF mask prediction as a supervised learning problem — choice of target TF mask, features and the learning algorithm.
- Use of recurrent neural networks — specifically long short term memory (LSTM) networks — for predicting TF masks.
- Assessment of post-filtered speech in terms of objective speech intelligibility and quality measures, followed by interpretation of those scores against the observed word error rates of the then state-of-the-art ASR back-end.

The organization of this thesis is as follows. Chapter 2 throws some light on the diverse topics discussed in this thesis — Figure 1.2 illustrates this diversity. The methods used in the proposed framework are discussed in Chapter 3. In Chapter 4, the experimental set-up and evaluation of our methods on CHiME-3 Challenge data are presented followed by results and discussion. Conclusions are derived and future directions are identified in the final chapter.

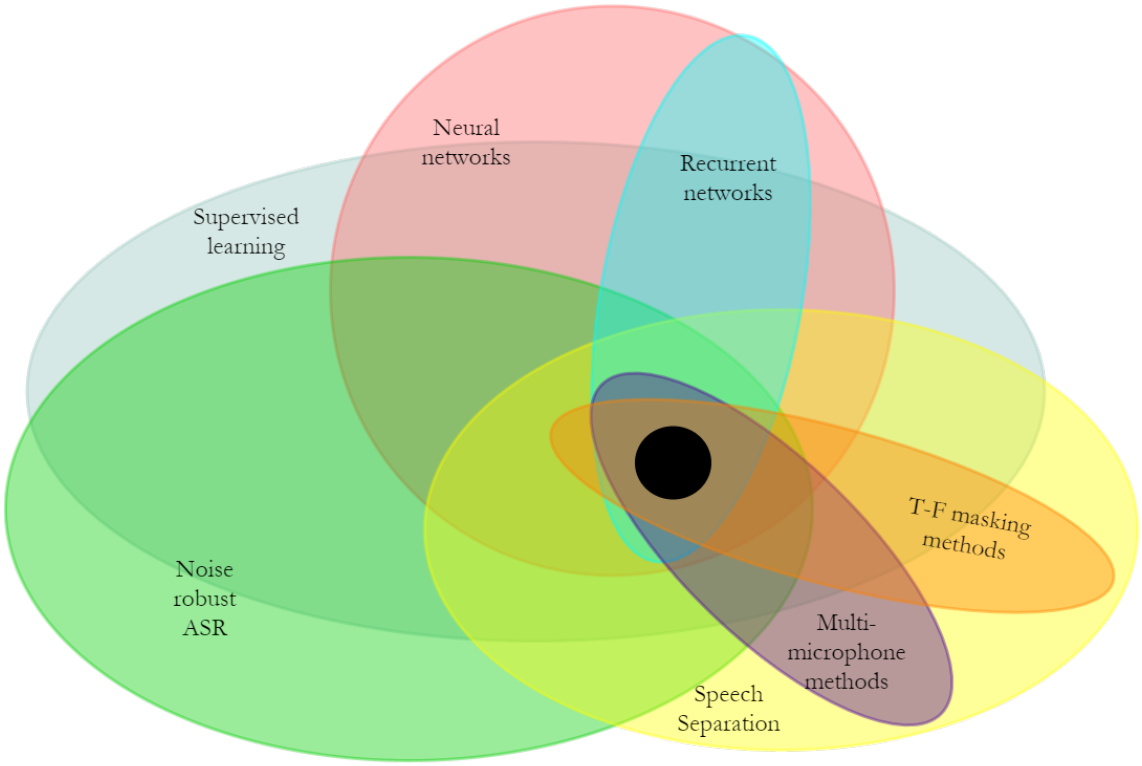


Figure 1.2 *Depicting the relationship among the topics discussed in this thesis.*

2. BACKGROUND

This chapter splits open every topic as illustrated in Figure 1.2 and serves as a primer for them. Sections 2.1 – 2.4 introduce the fundamental problem, the inherent difficulties in addressing it and other terminologies associated with the topics discussed later. Section 2.5 introduces microphone array processing and beamforming technique. Section 2.6 gives a general overview of supervised learning using neural networks, focusing on recurrent networks and long short-term memory architecture. Sections 2.7 – 2.9 presents the application of supervised learning to speech separation, detailed literature review and concepts related to automatic speech recognition.

2.1 On time-frequency representations

The time-domain signal model describing a single-channel audio recording of a stationary sound source in an acoustic environment can be stated as

$$m(t) = s(t) * h(t) + v(t), \tag{2.1}$$

where $m(t)$ is the audio signal captured by the microphone, $s(t)$ is the source signal, $h(t)$ is the impulse response of the channel, $v(t)$ is the additive sensor noise, $*$ is the convolution operation and t is the time. In a multi-source environment, the signal model becomes

$$m(t) = \sum_{p=1}^{P} s_p(t) * h_p(t) + v(t), \tag{2.2}$$

where P is the total number of sources, $s_p(t)$ is the pth source signal, $h_p(t)$ is the impulse response of the channel corresponding to that source. $m(t)$ can now be called a *mixture* signal.

An audio signal in its raw pulse coded modulation (PCM) format represents the variation of acoustic pressure as a function of time. This *native* representation — the representation that is closest to the original analog signal — is often difficult to interpret. This leads to its frequency domain representation — though not completely eschewing other *derived* time domain representations such as autocorrelation function, zero-crossing rate, linear prediction coefficients — with the aid of

discrete Fourier transform (DFT). Many characteristic properties of speech signals such as vocal tract resonances — which characterize voiced sounds — can be easily visualized from its magnitude spectral envelopes. But when DFT is applied to the entire time-limited signal — for example that covers an entire sentence — the resulting *average* spectrum does not convey information such as the evolution of fundamental frequencies over time. This can be circumvented by analyzing the signals over small segments or windows and obtaining the time evolution of spectrum or a time-frequency (TF) representation.

The fundamental and most frequently used TF representation is obtained by computing short-time Fourier transform (STFT) and the resulting representation is called a *spectrogram*. Computing STFT involves dividing the discrete-time signal into overlapping — the amount of overlap determined by the *hop-size* — or non-overlapping frames, multiplying them with a suitable window function and finally computing the DFT of the windowed frames. If the short frames $m_n(t)$ of the signal $m(t)$ are obtained after windowing with w_{STFT}, then the STFT can be expressed as

$$m(n,k) \equiv \mathcal{F}\big\{m_n(t)\big\} = \mathcal{F}\big\{m(t)w_{\text{STFT}}(t-nR)\big\}, \tag{2.3}$$

where $t, n, k \in \mathbb{Z}$ are the sample index, frame index and frequency index respectively, R is the hop-size in samples and $\mathcal{F}$ is the DFT operator. The above description of STFT is called *overlap-add* (OLA) interpretation of STFT wherein STFT is seen as a *time-ordered* sequence of spectra. The window functions should obey the constant overlap-add (COLA) property [6], which when mathematically put,

$$\sum_{n\in\mathbb{Z}} w_{\text{STFT}}(t-nR) = 1, \forall k, t \in \mathbb{Z}. \tag{2.4}$$

The COLA property is the direct consequence of the requirement that $m(t) = \sum_{n\in\mathbb{Z}} m_n(t)$. The rectangular window with 0 or 50% overlap, Bartlett window with a 50% overlap, any member of generalized Hamming family with a 50 % overlap, any member of Blackmann family at 2/3 overlap are examples of window functions that satisfy COLA property [6].

STFT has another dual interpretation as a *filter-bank* wherein it is seen as a *frequency-ordered* sequence of narrow-band time-domain signals — each *channel* of the filter-bank is equivalent to low-pass filtering a *heterodyned* signal [6]. On parallel lines, for a perfect reconstruction — similar to COLA property of analysis windows — the frequency responses of the individual channels overlap-add to a constant over the unit circle.

There exists the *uncertainty* principle for STFT based TF representations, according to which both time and frequency localization cannot be achieved simultaneously — one has to be traded-off for the other. Narrow-band spectrograms are more accurate in frequency dimension and used in applications like assessing the vowel intonation — accomplished by resolving individual harmonics of the voiced source. Wide-band spectrograms on the other hand emphasize temporal variations in the signal and can be used, for example, in reliably obtaining information about the timing of changes in vocal tract resonance.

By invoking the perceptual motivation of mimicking the human auditory system — that human perception of pitch is roughly linear below 1 kHz and logarithmic above — a *Hertz* to *mel* scale transformation can be achieved as

$$\mathrm{MEL}(f_{\mathrm{Hz}}) = 2595 \ \log_{10}\left(1 + \frac{f_{\mathrm{Hz}}}{700}\right) \tag{2.5}$$

The above transformation is typically accomplished with the aid of a filterbank consisting of triangular filters that are equally spaced along the mel scale — in a linear scale, this is equivalent to having narrow-band filters in low frequency regions with the bandwidth of subsequent filters progressively widening as we traverse the frequency scale. The resulting TF representation is called a *mel spectrogram*, the number of mel bands deciding its resolution. Since human perception of sound intensity is logarithmic, it is also judicious to compute the logarithm of the mel *energies* resulting in a *log mel spectrogram*. Instead of *mel* filterbank, another perceptually motivated filterbank called *gammatone* filterbank can also be used — the resulting representation is often attributed as a *cochleagram*.

2.2 Separation or enhancement? — differing notions

The possible solutions to the cocktail party problem comprise of speech separation, speech enhancement and noise reduction — all of them intimately tied to each other and together aim to improve speech quality and/or intelligibility (the terms quality and intelligibility are explained in the following subsection); but their definitions usually differ based on the context. This section will present those differing definitions and my appropriate stance on adopting "separation" instead of the popular notion of "enhancement".

For example, according to [7], speech enhancement refers to the problem of recovering the target speaker's speech from stationary or nearly stationary backgrounds containing, for example, car noise, babble-like and non-speech-like sources.

Speech separation, on the other hand, refers to the case where the background is non-stationary, consisting of sources such as competing speakers or music [7]. A similar view is also supported by [8] where they define speech separation or *segregation* as the general task of separating the target speech from background interference — non-speech, interfering speech or both — as well as reverberation. Speech enhancement or *denoising* means separation of speech and non-speech noise. I will be sticking to this viewpoint throughout this thesis.

In popular view, if the target source is fixed, irrespective of the nature of interference, it is a speech enhancement problem. Speech separation then implies the situation of multiple talkers speaking simultaneously and the goal being to extract the signals from all individual speakers — thus its meaning is derived from classical source separation problem; [8] defines this problem as *speaker* separation. Nevertheless, the methods encompassing the terminologies are solutions for cocktail party problem.

Numerous algorithms have been developed in the past decade for speech enhancement: *spectral subtraction*, in which an estimate of short-term noise power spectrum is subtracted from the mixture spectrum to produce an estimate of clean speech spectrum; *Wiener filters* and other *minimum mean squared error* (MMSE) based approaches which are optimal in mean squared error (MSE) sense, in that, they minimize the squared error between the enhanced and clean signals.

Speech separation has been addressed by: *model-based* methods such as hidden Markov models (HMMs), non-negative matrix factorization (NMF) and independent component analysis; *spatial filtering* by using microphone arrays; *time-frequency (TF) masking* based methods; combination of the above [9]. In particular, TF masking based methods have tasted more success in separating under-determined (number of channels lesser than the number of sources) mixtures, especially that involve a single microphone. This is due to the fact that TF masking is essentially a filter with a time-varying magnitude response and by careful construction, they are more effective in combating interference. The principle behind TF masking and the algorithm is described in detail in Section 2.4

2.3 Speech quality vs intelligibility — how are they different?

The speech separation/enhancement systems are often independently assessed by two closely related — yet not equivalent — attributes of speech, viz., speech *intelligibility* and speech *quality* [10]. In the ISO 9921 standard, speech intelligibility is defined as "a measure of effectiveness of understanding speech" and plays a major

role in the evaluation of hearing aids and cochlear implants. Speech intelligibility focuses on what has been spoken — how much of the speech is correctly perceived and recognized. It decreases in the presence of background noise and reverberation, with stationary and non-stationary noises having different effects on it. Speech quality, on the other hand, is more subjective and assesses how much clear, pleasant and natural the utterance sounds and how much distortion-free it is.

The relationship between quality and intelligibility is not entirely understood — an utterance with poor quality can be more intelligible and vice versa; methods that improve one need not improve the other. For instance, spectral subtraction and Wiener filter based speech enhancement methods have been shown to improve speech intelligibility but not necessarily quality — the hypothesized reasons are as follows: the background noise spectrum can never be accurately estimated, especially for non-stationary noises; hence speech enhancement algorithms come along with induced artifacts called *musical noise* — so called because the isolated noise energy peaks that remain after enhancement are perceived as time-varying tones [11]; in an attempt to suppress the musical noise, the enhancement algorithms come along with techniques to improve the overall quality but with a compromise for intelligibility. Moreover, most of the existing algorithms optimize a cost function — for e.g., Wiener filtering employs mean squared error (MSE) between clean and estimated spectra to produce the best output — which does not necessarily correlate with speech intelligibility.

2.4 On time-frequency masking

In their high-resolution TF representation, the sound sources comprising a mixture are typically assumed to be *sparse*; in other words, most of the *bins* in their TF representation contain very low or zero energy. In addition, the individual sources — for a given windowing function $w(t)$ — are assumed to be approximately *w-disjoint orthogonal* [12]. In simple terms, the non-zero TF bins corresponding to each source do not mostly overlap. This orthogonality can be mathematically expressed as

$$s_i(n,k)\ s_j(n,k) = 0,\ i \neq j, \forall n, k, \tag{2.6}$$

where $s_i(n,k)$ and $s_j(n,k)$ are the TF representations of any individual sources.

The above assumptions of sparseness and disjointness form the basis for discriminating the TF bins of the mixture between the constituent sources based on *TF masking* (not to be confused with the masking phenomenon that is encountered

in psycho acoustics). Traditionally TF masking has been applied for reduction of noise, reverberation and interference by multiplying the observed noisy magnitude spectrogram with a real-valued mask, the key idea being applying low weights to TF-regions dominated by noise and leaving the target signal unchanged. For example, a binary TF mask can be identified and formed in which the bins corresponding to the portions of the mixture where the target sound is stronger than background are assigned a value of 1 and rest to 0. The *ideal binary mask*(IBM) can be formally defined as

$$y_{\text{IBM}}(n,k) = \begin{cases} 1, & \text{if SNR}(n,k) > \text{LC} \\ 0, & \text{otherwise,} \end{cases}, \tag{2.7}$$

where $\text{SNR}(n,k)$ is the local SNR of the mixture at the TF unit (n,k), LC is a local criterion, whose choice has a strong impact on speech intelligibility and is typically set to be 5 dB smaller than the mixture SNR [13].

This mask can then be multiplied with the TF representation of the mixture to yield an estimate of the desired source; the mask acts as a time-varying filter by suppressing the TF regions containing noise and interference while passing the regions corresponding to the signal of interest. IBM has already been suggested as a goal for CASA (computational auditory scene analysis) systems [14] — machine systems that are capable of achieving a human-level performance in auditory scene analysis (ASA) [15, 16]. Though IBM processing can yield large improvements in intelligibility even under low SNR conditions [17, 18], the quality of separated speech is a persistent issue because of the musical artifacts, introduced here due to the binary nature of the mask.

In practice, sources are not strictly sparse nor disjoint and a more plausible approach in such case is to use a *soft* mask, where the bins corresponding to a particular source are "close to" one and the rest to zero. The classical Wiener filter can be seen as an example of a soft TF mask where every TF bin of the mask is the ratio of target speech energy to the noisy mixture energy. Wiener filter is defined as

$$y_{\text{Wiener}}(n,k) = \frac{|s(n,k)|^2}{|s(n,k)|^2 \; + \; |v(n,k)|^2} \tag{2.8}$$

where $s(n,k)$ is the spectrogram of the target source and $v(n,k)$ is the spectrogram of the background.

Wiener filter is typically seen as a special case of the more general *ideal ratio*

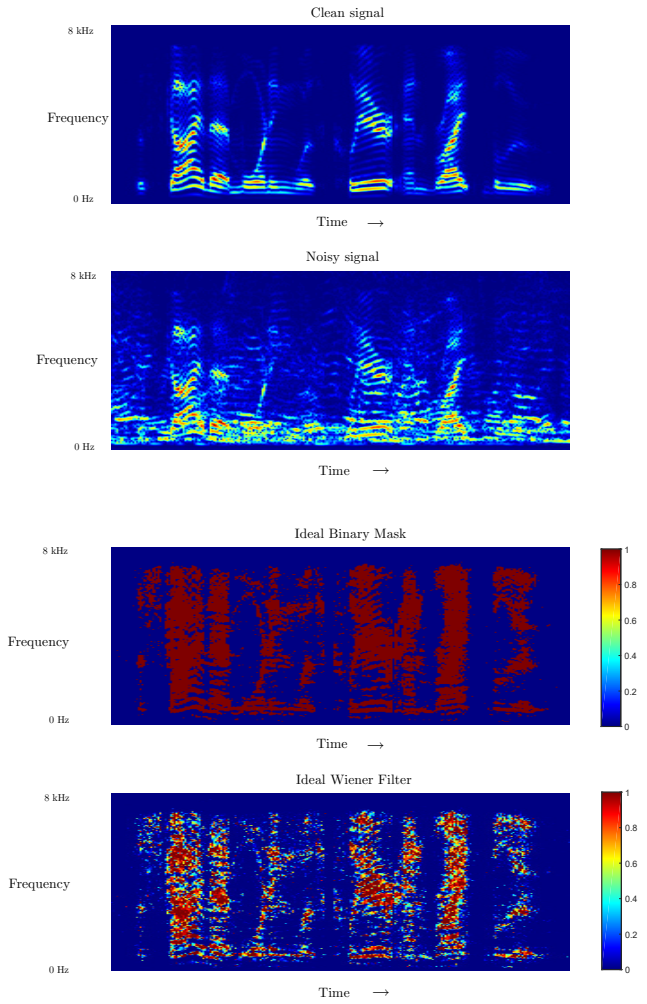


Figure 2.1 *A babble noise is added to the original clean signal at SNR=0 dB. The IBM with a local criterion of −5 dB and the Wiener filter are illustrated. The challenge of CASA is to estimate these masks when the clean signals are unavailable.*

mask (IRM) defined as,

$$y_{\text{IRM}}(n,k) = \left(\frac{|s(n,k)|^2}{|s(n,k)|^2 \ + \ |v(n,k)|^2} \right)^{\beta}, \tag{2.9}$$

where β is a tunable parameter. IRM has been observed by [19] to be more closely related to auditory processes than IBM, as ascertained by certain speech intelligibility and ASR measurements. It has also been observed as a goal of CASA recently [20]. An example visualization of IBM and Wiener masks for an utterance is given in Figure 2.1.

The maximum obtainable speech intelligibility is dictated by an *oracle* mask and the computation of such masks requires both the mixture and the corresponding clean signals, but the latter is not available in reality; so the masks have to be estimated. Though the perceptual reasons for the choice of a TF mask — is it binary, ratio, Wiener or some other? — are still unclear, the methods used in estimating the masks are found to play a significant role. Later in section 2.7.2, these TF masks appear as targets for supervised speech separation.

2.5 Microphone arrays

This section introduces microphone array terminologies — the signal model, techniques and the theoretical bases of methods used.

2.5.1 Delay-and-sum beamforming and time difference of arrival estimation

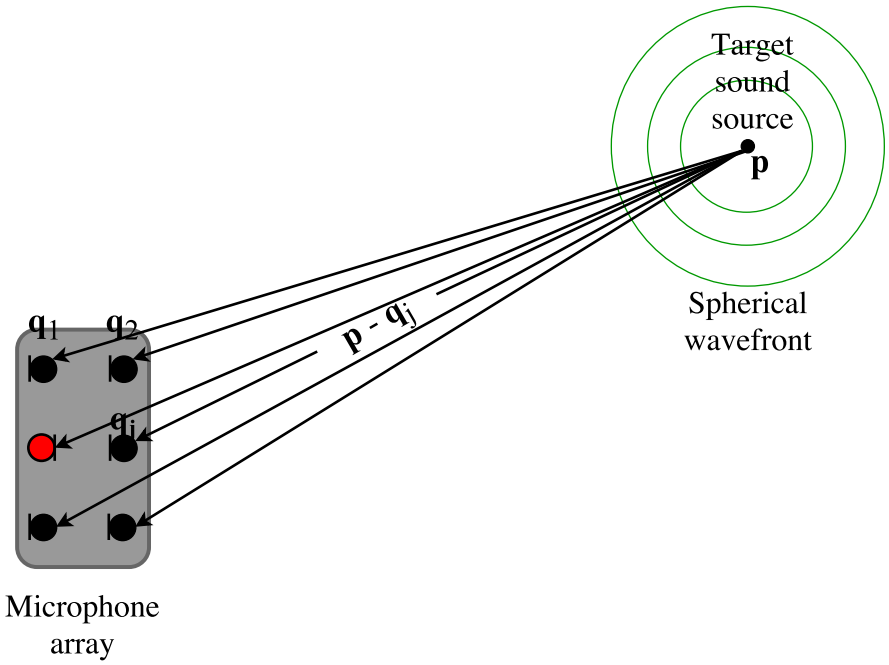


Figure 2.2 *Microphone array delay model*

Shown in Figure 2.2 is a sound wave propagating from the source in location $\mathbf{p} \in \mathbb{R}^3$ (in Cartesian coordinates) to each microphone located at $\mathbf{q}_j \in \mathbb{R}^3$ for $j = [1, ..., J]$, J is the total number of microphones in the array. A sound source can be "localized" by estimating source's position from time delay of arrival (TDoA) of the signals at

the microphones. Under the above setup, the propagation delay at jth microphone can be expressed as

$$\tau_d{}^{(j)} = \frac{||\mathbf{p} - \mathbf{q}_j||}{c}, \tag{2.10}$$

where $\tau_d{}^{(j)}$ is the propagation delay measured in seconds, c is the speed of sound in air and $||.||$ denotes the l_2 vector norm. Thus assuming that the signal at jth microphone is modeled as delayed and attenuated source signal with an additive noise and on the application of STFT, the signal model becomes

$$m_j(n,k) = \alpha_j \cdot s(n,k) \cdot \theta_j(n,k) + v_j(n,k), \tag{2.11}$$

where $s(n,k)$ is the source signal, $v_j(n,k)$ is the noise signal at microphone j which is independent and identically distributed from other microphones, α_j are the attenuation factors in the range between 0 and 1 due to propagation effects. The j^{th} component of array steering vector $\boldsymbol{\theta}$ is $\theta_j(n,k) \equiv exp(-\jmath\frac{2\pi k\tau_d^{(j)}F_s}{N_{DFT}})$, where $\jmath = \sqrt{-1}$, F_s is the sampling frequency and N_{DFT} is the DFT size.

The *delay and sum beamforming* (DSB) is the most fundamental beamforming technique. In theory, the DSB combines the outputs of J microphone signals as

$$m_{DSB}(n,k) = \sum_{j=1}^{J} w_{DSB}^{(j)}(n,k) \cdot m_j(n,k), \tag{2.12}$$

where $w_{DSB}^{(j)}(n,k) \equiv \theta_j^*(n,k) = exp(\jmath\frac{2\pi k\tau_d^{(j)}}{N_{DFT}})$ are the beamformer weights and $(.)^*$ is the complex conjugation operation. This way, the individual microphone signals are time-aligned before summation to obtain an enhanced signal.

For signal enhancement, the absolute propagation delays $\tau_d^{(j)}$ need not be known; rather the *time difference of arrival* (TDoA) for a microphone pair $\{i,j\}$ defined as $\Delta\tau_d^{(i,j)} \equiv \tau_d^{(i)} - \tau_d^{(j)}$, j being the reference microphone, can be calculated. For $i = j,\ \Delta\tau_d^{(i,j)} = 0$. Thus DSB output can be written as

$$m_{DSB}(n,k) = \sum_{j=1}^{J} exp\left(\jmath\frac{2\pi k\Delta\tau_d^{(i,j)}}{N_{DFT}}\right) \cdot m_j(n,k). \tag{2.13}$$

There are various established techniques for estimating the TDoAs. The TDoA estimation is not the primary focus of this thesis; a good overview is provided in [21]. The widely used *generalized cross-correlation using phase-transform* (GCC-PHAT) method [22] is succinctly presented below. For brevity and to be consistent with

other literature continuous-time notations are used.

Time delay between two microphone signals can be estimated by calculating cross-correlation between two signals as

$$r_{m_i,m_j}(\tau) = \int_{-\infty}^{\infty} m_i(t) \cdot m_j(t+\tau)\ dt, \tag{2.14}$$

where $r_{m_i,m_j}(\tau)$ is the cross-correlation function, $i \neq j$ and τ is the time lag. Also by cross-correlation theorem,

$$r_{m_i,m_j}(\tau) = \mathcal{F}^{-1}\{M_i(f) \cdot M_j(f)^*\}, \tag{2.15}$$

where $\mathcal{F}$ is the Fourier transform operator, $\mathcal{F}^{-1}$ is the inverse Fourier transform operator, $*$ indicates complex conjugation operation and $M_i(f) = \mathcal{F}\{m_i(t)\},\ M_j(f) = \mathcal{F}\{m_j(t)\}$ respectively. The time delay between m_i and m_j can be estimated by seeking a τ that maximizes the cross-correlation:

$$\Delta\tau_d^{(i,j)} = \arg\max_{\tau}\ r_{m_i,m_j}(\tau) \tag{2.16}$$

Estimating TDoA thus leads to locating peaks in the cross-correlation function. In reality, due to the presence of background noise and reverberation, spurious peaks appear and render the problem more challenging. The generalized cross-correlation (GCC) improves the robustness of TDoA estimation by introducing weighs to the cross-correlation function. For example, using phase transform (PHAT), a popular weighing scheme, TDoA estimation problem becomes

$$\Delta\tau_d^{(i,j)} = \arg\max_{\tau}\ r_{m_i,m_j}(\tau) \equiv \mathcal{F}^{-1}\Big\{\frac{M_i(f) \cdot M_j(f)^*}{|M_i(f) \cdot M_j(f)^*|}\Big\}. \tag{2.17}$$

Assuming that the noise signals at individual microphones have the same energy and the attenuation factors α_n equal to unity, the output SNR of a DSB beamformer can be written as [21]

$$\mathrm{SNR}_{out} = \frac{J}{1+\rho_s} \cdot \mathrm{SNR}_{in}, \tag{2.18}$$

where

$$\rho_s = \frac{2}{J}\sum_{i=1}^{J-1}\sum_{j=i+1}^{J} \rho_{v_i v_j} \tag{2.19}$$

$$\rho_{v_i v_j} = \frac{E(v_i(t)v_j(t+\tau))}{\sigma_{v_i}\ .\ \sigma_{v_j}}, \tag{2.20}$$

where SNR_{out} is the signal-to-noise ratio (SNR) of the output of the beamformer and SNR_{in} is the input SNR, $\rho_{v_i v_j}$ is the correlation coefficient, with $|\rho_{v_i v_j}| \leq 1$, σ_{v_i} and σ_{v_i} are the noise standard deviations. Thus, if the noise signals at the microphones are uncorrelated, a simple time-shifting and adding operation among the sensor outputs yield an SNR improvement by a factor equal to the number of sensors.

DSB belongs to a class of beamformers called *fixed beamformers*. DSBs have limited ability in suppressing noise and competing sources, as it typically requires a larger array to achieve sharper *directivity* [23]. In other words, the array *beamwidth* cannot be decreased unless the spacing between the individual microphones in the array is increased — but not any greater than $\lambda/2 = c/(2f)$, λ being the wavelength of the signal, as it may lead to *spatial aliasing* [21]. This is an important issue in applications such as automatic speech recognition, hearing aid and mobile communication which typically depend on small-sized arrays.

2.5.2 Adaptive beamforming

Adaptive beamforming techniques such as MMSE, minimum variance distortionless response (MVDR) and the broader class of linearly constrained minimum variance (LCMV) response based methods use the characteristics of both the source and noise signals to achieve better SNR gain. The beamformer weights are typically derived by constrained optimization — for example, the weights for an MVDR beamformer are chosen by minimizing the output power of the beamformer with the constraint that the desired signal is not distorted. This is also equivalent to minimizing the output noise power while maintaining the energy along the target direction. The MVDR problem is thus the following quadratic optimization problem [21]

$$\mathbf{w}_{MVDR} = \arg\min_{\mathbf{w}} \left\{\mathbf{w}^T \mathbf{R}_v \mathbf{w}\right\} \text{ subject to } \mathbf{w}^T \theta = 1, \tag{2.21}$$

resulting in

$$\mathbf{w}_{MVDR} = \frac{\mathbf{R}_v^{-1}\theta}{\theta^H \mathbf{R}_v^{-1}\theta}, \tag{2.22}$$

where $\mathbf{R}_v$ is the spatial covariance of the noise and θ is the array steering response vector. Accurate estimation of $\mathbf{R}_v$ and θ are thus crucial for MVDR.

The MVDR offers no distortion to the signal components that are protected by the constraints but do not sufficiently suppress the noise. Alternatively, in an MMSE approach, the system is designed to minimize the mean-squared-error w.r.t

the desired components and results in better noise reduction at the cost of distorting the desired signal. Adaptive beamformers are studied in detail in [21].

2.5.3 Post-filtering

A stand-alone beamformer cannot simultaneously handle diffuse-noise suppression and leave the desired signal undistorted. In order to obtain the MMSE estimate of the desired signal components, a Wiener filter is typically applied at the output of non-MMSE type beamformers [24]. Such Wiener filters are called *post-filters*.

Design of such post-filters are dependent on the beamformer type, uses many assumptions and typically involves estimation of power spectrum from the observed signals containing noise. For example, Zelinski post-filter [25], using the auto- and cross- power spectra of the multi-channel input signals, estimates the target and noise power spectra, based on the assumption of zero cross-correlation between the noises at different microphones. The assumption is valid only for larger-sized arrays when the distances between microphones are large enough compared to the wavelength. Inaccurate estimation of spectra thus results in musical artifacts leading to poor intelligibility.

In the quest to address the above list of problems for a small-sized, mobile microphone array, the use of post-filter based on TF mask estimation using supervised learning techniques is explored.

2.6 Supervised learning and neural networks

As previously mentioned, we are interested in the problem of "estimating" TF masks for speech separation by data-driven approaches such as supervised learning. This section gives a basic introduction to the supervised learning framework and associated terminologies.

2.6.1 The learning problem, feasibility of learning and generalization aspects

In any supervised learning problem [26], there is an input $\mathbf{x} \in \mathbb{R}^N$ and an unknown *target function* $G : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X}$ is the space of all possible $\mathbf{x}$ (input space) and $\mathcal{Y}$ is the output space — a space of all possible $\mathbf{y} \in \mathbb{R}^M$. Thus $\mathbf{y} = \mathrm{G}(\mathbf{x})$. In a supervised setting, there is a *dataset* $\mathcal{D}$ of input-output examples $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), ..., (\mathbf{x}_N, \mathbf{y}_N)\}$, N is the size of the dataset. Finally, there is the *learning algorithm* that uses $\mathcal{D}$ to choose a *hypothesis* $H : \mathcal{X} \rightarrow \mathcal{Y}$, $H \in \mathcal{H}$ that

approximates G. $\mathcal{H}$ is the *hypothesis* set, for example, it can be a set of all linear functions. In other words, the learning algorithm infers H given the dataset $\mathcal{D}$. For any datapoint $\mathbf{x}$, the output can thus be predicted as $\hat{\mathbf{y}} = H(\mathbf{x})$.

In addition to the above, a learning problem also requires an *error measure* that quantifies how well H approximates G. For regression problems, mean squared error can be an example of the error measure defined as

$$E_{in}(G, H) = \frac{1}{N}\sum_{n=1}^{N} ||\mathbf{y}_n - \hat{\mathbf{y}}_n||^2 \equiv \frac{1}{N}\sum_{n=1}^{N} ||G(\mathbf{x}_n) - H(\mathbf{x}_n)||^2, \tag{2.23}$$

where $E_{in}(G, H)$ is the "in-sample" error measure and refers to error computed for data samples in the dataset $\mathcal{D}$. In this context, $\mathcal{D}$ is called the training dataset.

The goal of any learning problem is to perform well on unseen data and this is evaluated by "out-of-sample" error $E_{out}(G, H)$ defined similarly as in equation 5.1, but for data points outside $\mathcal{D}$. Learning is feasible if

1. $E_{out}(G, H) \approx E_{in}(G, H)$,
2. $E_{in}(G, H) \approx 0$

While the second aspect is a direct consequence of G approximating H, the first aspect is quite subtle and influenced by the complexity of the model. If there is a huge difference between E_{out} and E_{in}, it means that the model is not generalizing well, in other words, the "complex" model is *overfitting* the dataset $\mathcal{D}$. On the other hand, if the second aspect is not met, the model is *underfitting* the dataset. E_{in} is always tractable while training and hence the second aspect is not a major issue in the context of learning.

To address the generalization aspect, typically, the original dataset $\mathcal{D}$ is split into 3 disjoint sets:

Training set $\mathcal{D}_{tr}$: used in learning the parameters of the model,

Validation set $\mathcal{D}_{val}$: used in monitoring overfitting (also used in choosing the hyper-parameters of a learning model which is discussed later),

Test set $\mathcal{D}_{test}$: used in reporting the final performance.

Care should be taken that $\{\mathcal{D}_{tr} \cap \mathcal{D}_{val} \cap \mathcal{D}_{test}\} = \emptyset$. To avoid overfitting, proper regularization techniques should be used, the techniques being dependent on the learning models used.

In reality, additional challenge comes in the form of noisy targets, resulting in learning a *target distribution* $P(y|\mathbf{x})$ instead of a target function $y = G(\mathbf{x}) = \mathbb{E}(y|\mathbf{x})$.

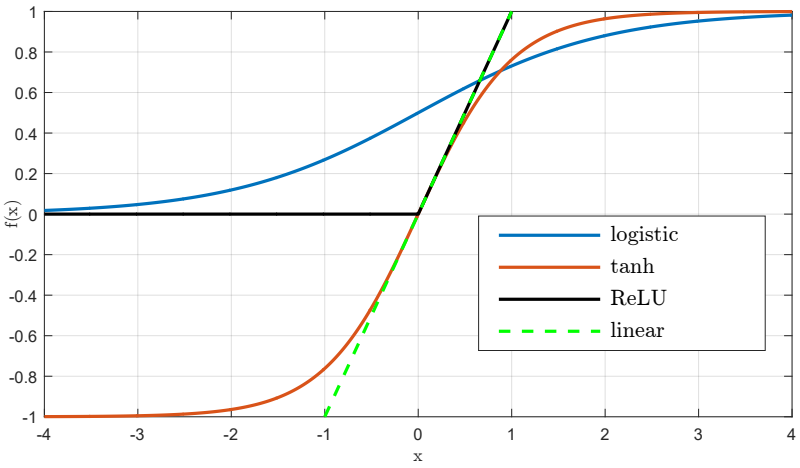


Figure 2.3 *Common activations used in neural networks*

2.6.2 Feed-forward neural networks

Artificial neural networks — by learning theory principles — form a complex, non-linear hypothesis set and have a long history of successful applications in pattern classification, non-linear regression and time-series prediction problems. A neural network is made up of a set of interconnected fundamental building blocks called neurons. A single neuron is modeled as a non-linear function of the weighted linear combination of its inputs and a bias term as

$$o = f(\mathbf{w}^T\mathbf{p} + b), \tag{2.24}$$

where $o \in \mathbb{R}$ is the output of the neuron, $f(\ .\ )$ is the non-linear function called the activation function, $\mathbf{w}, \mathbf{p} \in \mathbb{R}^{N_i}$ are the weight vector of the neuron and input vector to the neuron respectively, N_i is the input vector dimension and $b \in \mathbb{R}$ is the bias term. The activation function is a differentiable function, usually of "squashing" type such as the *logistic* (*sigmoid*) function, *hyperbolic tangent* function or *rectified linear unit* (ReLU) and are illustrated in Figure 2.3 .

Neurons in a neural network are arranged in layers — an input layer, followed by a variable number of hidden layers and an output layer. The sizes of the input and output layer are the feature vector and target vector dimensions, respectively. If the outputs of preceding layer neurons solely form the inputs of the succeeding layer neurons, the network is called a feed-forward neural network (FFNN) and often called a multi-layer perceptron (MLP). It has to be noted that other sophisticated *convolutional neural networks* (CNNs) also belong to the feed-forward type, though

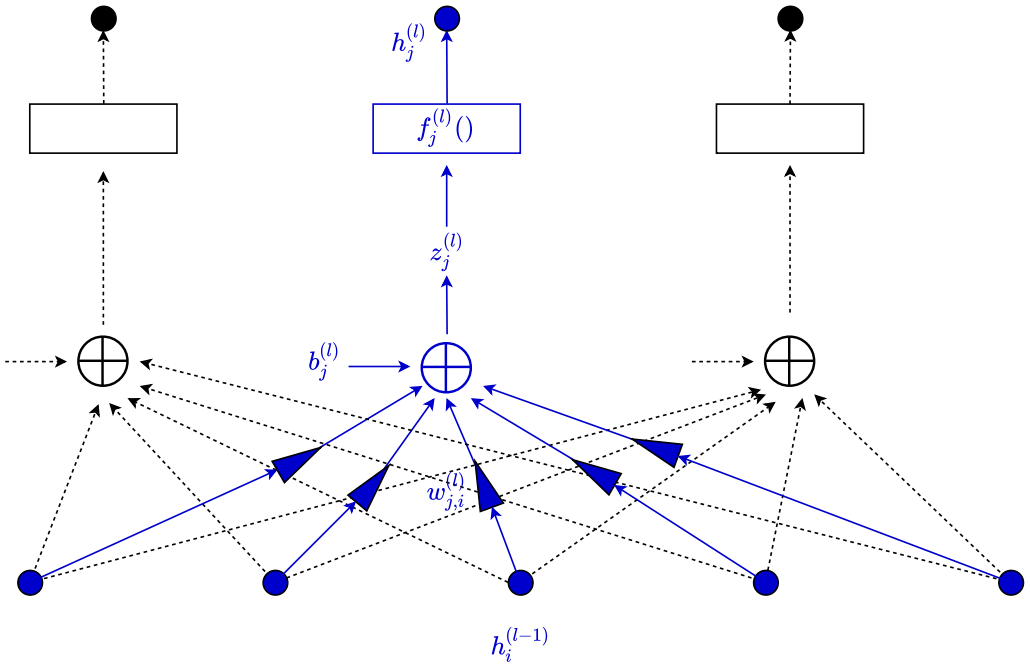


Figure 2.4 *A single layer of neural network illustrated. Highlighted in blue is a single neuron.*

MLPs are often attributed as FFNNs. Thus in an FFNN, the output vector of a single layer can be expressed as (also refer to Figure 2.4)

$$\mathbf{h}^{(l)} = f(\mathbf{z}^{(l)}) = f(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}), \tag{2.25}$$

where l is the current layer index, $\mathbf{W}^{(l)} \in \mathbb{R}^{N_l X N_{l-1}}$ is now the weight matrix, N_l is the current hidden layer size, N_{l-1} is the previous hidden layer size (also the input dimension of the current layer). $\mathbf{h}^{(l)}, \mathbf{b}^{(l)} \in \mathbb{R}^{N_l}$ are the output vector & bias vector of the current layer respectively. $\mathbf{z}^{(l)}$ — often attributed as *net input* — is the summed output prior to activation. A hidden layer is said to be fully-connected to the previous layer if every output of previous layer is connected to every neuron of the current layer and in such case, $\mathbf{W}^{(l)}$ will be a dense matrix.

The hidden layer weight values and the biases form the parameters of the learning model. The choices of the number of hidden layers, hidden layer size and the activation function used, together form the "hyperparameters" of the neural network model. The complexity of the model and its generalization capability depend on these hyperparameters.

For regression problems, the output layer activation function is typically linear unlike other squashing functions used in the hidden layers; while for classification,

sigmoid or tanh are still used as output activations. It has to be pointed out that if linear activations are used in all the hidden layers, the MLP effectively performs only linear regression (or finds only a linear boundary in its classification counterpart). Any multi-layer perceptron with linear hidden activations is equivalent to some other single layer perceptron in the hypothesis space.

The main objective now is to solve the learning problem as described in Section 2.6.1 where the hypothesis set is now a set of MLPs. The MSE cost function to optimize — $\mathbf{E}(\mathbf{W}, \mathbf{B}, \mathbf{x_n}, \mathbf{y_n}) = \frac{1}{N}\sum_{n=1}^{N} ||\mathbf{y}_n - \hat{\mathbf{y}}_n||^2$ — thus depends on the network weights $\mathbf{W} \equiv \{\mathbf{W}^{(l)}\}_{l=1}^{l=L}$, biases $\mathbf{B} \equiv \{\mathbf{b}^{(l)}\}_{l=1}^{l=L}$ and the training data $\{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{n=N}$, with the free parameters being the weights and the biases. In a typical learning scenario, a *learning algorithm* such as *stochastic gradient descent* is used to learn the hidden layer weights and the biases. The *backpropagation* algorithm is an efficient technique to compute the gradient of cost function w.r.t. weights and biases of each layer by applying chain rule for derivatives. While explaining the entire backpropagation algorithm is a chapter on its own, Figure 2.5 tersely summarizes the whole algorithm.

Once the gradients of weights and biases are computed, the weights and biases are updated based on their update modes. When the gradient update is done after processing every input-output pair, it is called *stochastic* mode. On the other extreme, if the update is done once gradients are computed for all input-output pairs, it is called *full-batch* mode. While stochastic mode may not lead to true gradient but is faster in general, *full-batch* mode yields the most accurate gradient at the cost of large computation per update. Typically, *mini-batch* updates are done in which gradients are averaged over a mini-batch of samples and the update is done as

$$\mathbf{W}^{(l)} = \mathbf{W}^{(l)} - \frac{\eta}{N_{batch}} \sum_{(\mathbf{x},\mathbf{y})\in batch} \nabla_{\mathbf{W}^{(l)}} E, \tag{2.26}$$

$$\mathbf{b}^{(l)} = \mathbf{b}^{(l)} - \frac{\eta}{N_{batch}} \sum_{(\mathbf{x},\mathbf{y})\in batch} \nabla_{\mathbf{b}^{(l)}} E, \tag{2.27}$$

where η is a hyper parameter called *learning rate* which controls the learning speed and stability of convergence, N_{batch} is the *mini-batch size*, $\nabla_{\mathbf{W}^{(l)}} E$ and $\nabla_{\mathbf{b}^{(l)}} E$ are the gradients computed w.r.t lth layer weights and biases.

Basic stochastic descent leads to slow convergence and several variants have been proposed — adding weight *momentum*, *RMSprop*, *Adagrad* [27], *Adadelta* [28], *Nesterov accelarated gradient* (NAG), *Adam* [29] — which can speed up convergence and/or increase the optimization stability. The network training also involves

training over several passes of the data called *epochs*. For every epoch, the input training data can be randomly permuted and also different weight initialization can be

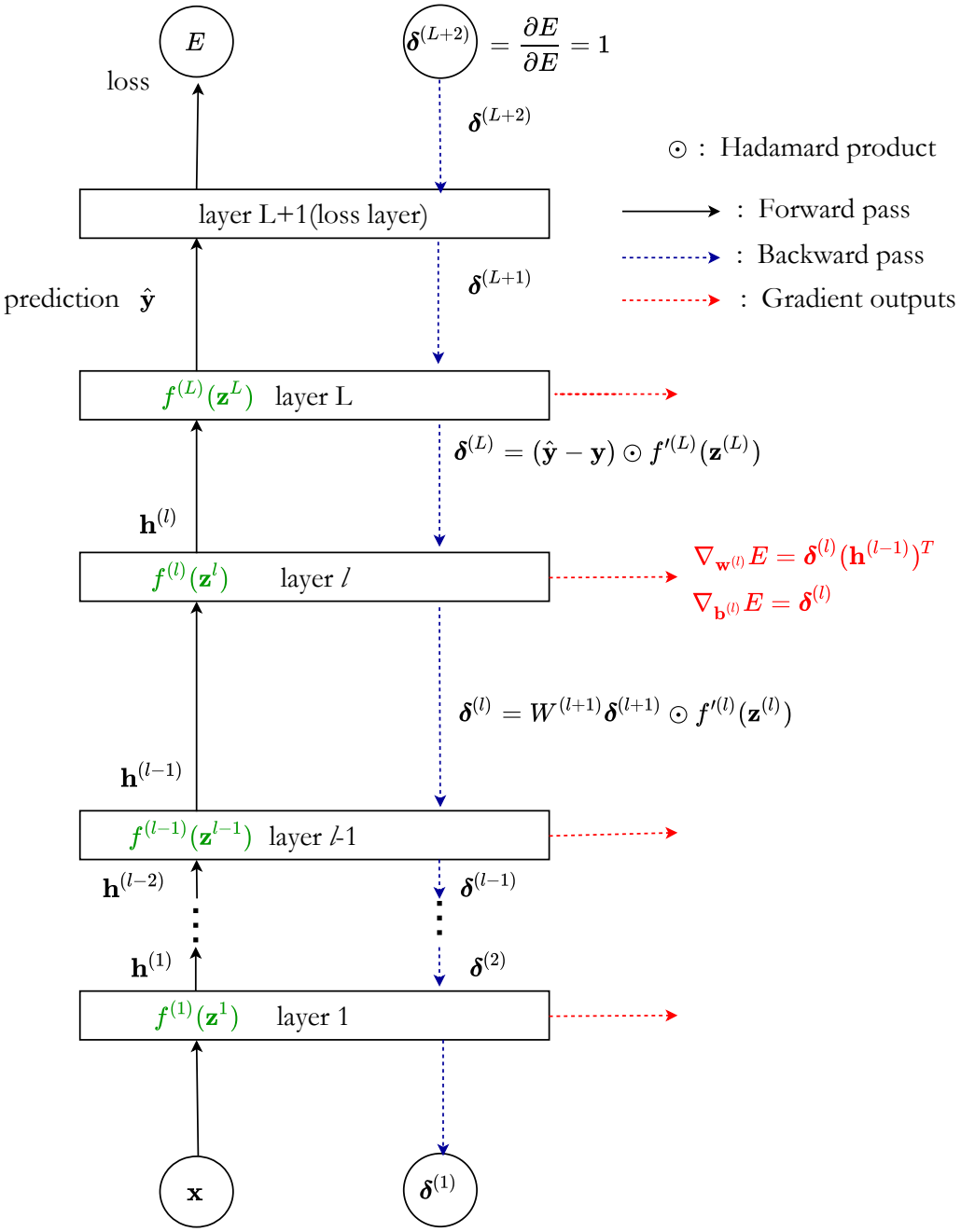


Figure 2.5 *The network is initialized with random weights and biases and the output is computed using the forward-pass equation 2.25. The overall loss and the backpropagated error terms $\delta^{(l)}$ for every layer are computed, which are then used to compute the gradients w.r.t. the layers' weights and biases*

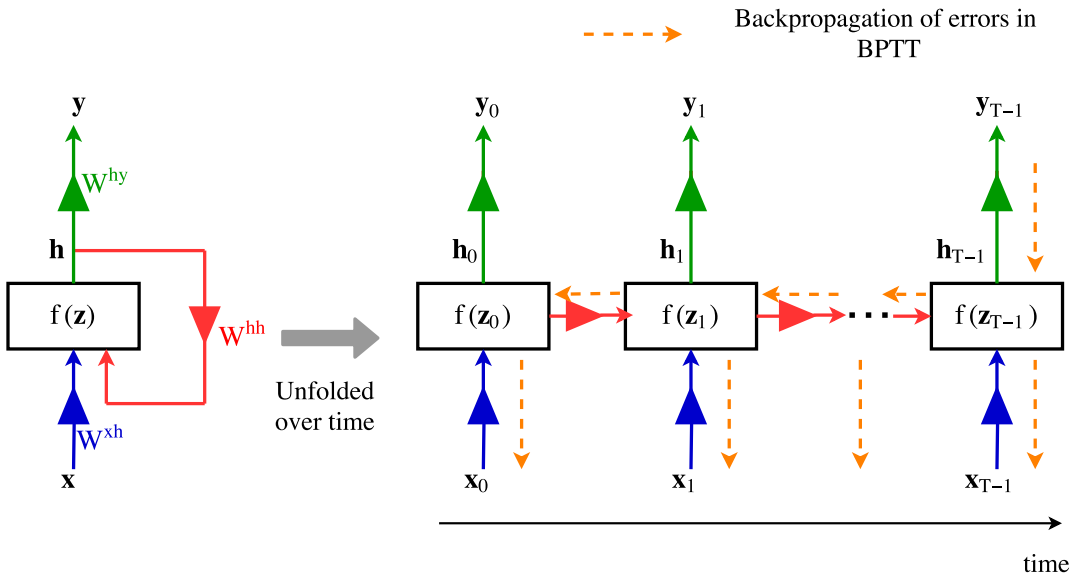


Figure 2.6 *A single layer RNN illustrated unfolded over time. For simplicity, assuming a linear output activation, output activation is omitted for clarity. The gradients to be computed for time-step $t = T - 1$ in BPTT is illustrated.*

The hyperparameters of the model are chosen with the aid of validation data — for every unique hyperparameter combination, the network is trained for several epochs across the training data using any of the gradient descent algorithms and for every epoch, the performance of the learned model on the validation data is monitored. The hyperparameter combination that yields the least validation error can then be finally chosen. The most simple way to choose hyperparameters in the hyperparameter space is by a grid-search — by exhaustively searching through a manually specified range of values for every hyperparameter. Another most commonly used technique is random search and covers the search space more uniformly.

2.6.3 Recurrent neural networks

An FFNN processes all samples independently of each other and hence cannot directly process sequential inputs unless the context information is artificially incorporated by concatenating past/future contextual features with the current feature. But this leads to increase in dimensionality leading to longer training time, larger models and hence also requires more data for better generalization. *Time-delay neural networks* (TDNN) which are designed to handle sequential inputs is one solution. But they are also limited in their "window size" and can thus accept only fixed amount of context.

Recurrent neural networks can also handle sequential inputs, albeit in a different manner compared to TDNN — RNNs use feedback connections in the hidden layers to keep track of past (and future if bi-directional RNNs are considered) context. This hidden layer recurrence makes RNNs more powerful as it can learn what to remember unlike TDNNs which are pre-wired to remember.

For a sequence of input vectors $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$, an RNN with single hidden layer computes a sequence of hidden activation vectors $\{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_T\}$ and a sequence of output vectors $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_T\}$ as

$$\mathbf{h}_t = f^h(\mathbf{W}^{xh}\mathbf{x}_t + \mathbf{W}^{hh}\mathbf{h}_{t-1} + \mathbf{b}^h), \tag{2.28}$$

$$\hat{\mathbf{y}}_t = f^{\hat{y}}(\mathbf{W}^{h\hat{y}}\mathbf{h}_t + \mathbf{b}^{\hat{y}}), \tag{2.29}$$

for all timesteps $t = 1, 2, ..., T$, where T is the sequence length, $\mathbf{W}^{xh}$, $\mathbf{W}^{hh}$ and $\mathbf{W}^{h\hat{y}}$ denote the weights connecting input-hidden, hidden-hidden (feedback connection) and hidden-output layers respectively, $\mathbf{b}^h$ and $\mathbf{b}^{\hat{y}}$ are the bias terms, and f^h and $f^{\hat{y}}$ are activation functions respectively.

In an RNN, information from previous time steps can virtually circulate indefinitely inside the network. The hidden activations thus create an internal state $\mathbf{h}_t$, for every hidden layer at each time-step. Equivalent to how MLPs can approximate any non-linear function, RNNs can approximate any dynamical system.

A straightforward extension of backpropagation algorithm called *backpropagation through time* (BPTT) can be used to train RNNs. To understand BPTT, a single layer RNN is shown unfolded over time in figure 2.6. The unfolded RNN can thus be seen as a deep FFNN with a layer for each time-step, except that weights are tied across time .

Unfortunately, as illustrated in the Figure 2.6, in the error back-propagation phase, the gradient signal is multiplied several times by the recurrent weight matrix $\mathbf{W}^{hh}$ since the output $\hat{\mathbf{y}}_t$ depends on the hidden state $\mathbf{h}_t$ which further depends on $\mathbf{h}_{t-1}...\mathbf{h}_0$. This results either in *vanishing gradients* when the product of the gradients vanish exponentially to zero or *exploding gradients* when the product explodes to infinity. This problem was first identified in [30] and discussed in detail in [31]. The exploding gradient problem can be attenuated by clipping the norms of the gradients, while the vanishing gradient problem has been historically difficult to solve, making long term dependencies difficult to learn. Out of several techniques to overcome the difficulties of training RNNs, complex architectures such as *long short-term memory* (LSTM)[32] and *gated recurrent units* (GRU) [33] have been the most successful.

The LSTM architecture is composed of a set of recurrently connected "subnets" called *memory blocks* as illustrated in Figure 2.7. Memory blocks allow an LSTM unit to adaptively forget, memorize and expose the memory content. Since its time of inception [34], LSTM memory blocks have gone through several modifications, such as addition of *forget gates* and introducing *peephole connections* [32]. The currently most widely used version will be discussed here. Each memory block consists of the following:

1. A self-connected *memory cell* $\mathbf{c}_t$ that stores the state — the new state is calculated by partially forgetting the existing memory content $\mathbf{c}_{t-1}$ and adding a new one $\bar{\mathbf{c}}_t$ ($\bar{\mathbf{c}}_t$ can be interpreted as candidates for the memory cell state which are further filtered by input decision gate),

2. *Input gate* $\mathbf{i}_t$ — controls the degree to which new memory content is added to the memory cell,

3. *Forget gate* $\mathbf{f}_t$ — resets the memory content if a feature is deemed unimportant, else the gate will be closed and the information is carried across many timesteps,

4. *Output gate* $\mathbf{o}_t$ — controls the amount of memory content to be yielded to the next hidden state

The existing memory content $\mathbf{c}_{t-1}$ are also shared to the gating neurons through peephole connections. The equations governing the LSTM-RNN operation are as follows:

$$\mathbf{i}_t = \sigma(\mathbf{W}^{\mathrm{xi}}\mathbf{x}_t + \mathbf{W}^{\mathrm{hi}}\mathbf{h}_{t-1} + \mathbf{W}^{\mathrm{ci}}\mathbf{c}_{t-1} + \mathbf{b}^{\mathrm{i}}), \tag{2.30}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}^{\mathrm{xf}}\mathbf{x}_t + \mathbf{W}^{\mathrm{hf}}\mathbf{h}_{t-1} + \mathbf{W}^{\mathrm{cf}}\mathbf{c}_{t-1} + \mathbf{b}^{\mathrm{f}}), \tag{2.31}$$

$$\bar{\mathbf{c}}_t = \tanh(\mathbf{W}^{\mathrm{xc}}\mathbf{x}_t + \mathbf{W}^{\mathrm{hc}}\mathbf{h}_{t-1} + \mathbf{b}^{\mathrm{c}}), \tag{2.32}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \bar{\mathbf{c}}_t, \tag{2.33}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}^{\mathrm{xo}}\mathbf{x}_t + \mathbf{W}^{\mathrm{ho}}\mathbf{h}_{t-1} + \mathbf{W}^{\mathrm{co}}\mathbf{c}_{t-1} + \mathbf{b}^{\mathrm{o}}), \tag{2.34}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{c}_t), \tag{2.35}$$

where σ is the logistic activation function, $\mathbf{W}^{\#\#}$ and $\mathbf{b}^{\#}$ are the weight matrices and bias terms respectively, $\odot$ is the element-wise product operation. The peepholes connections are not shared across memory blocks, in other words, the weight matrices $\mathbf{W}^{\mathrm{c}\#}$ are diagonal. A solid tutorial on RNNs and LSTMs is given in [35].

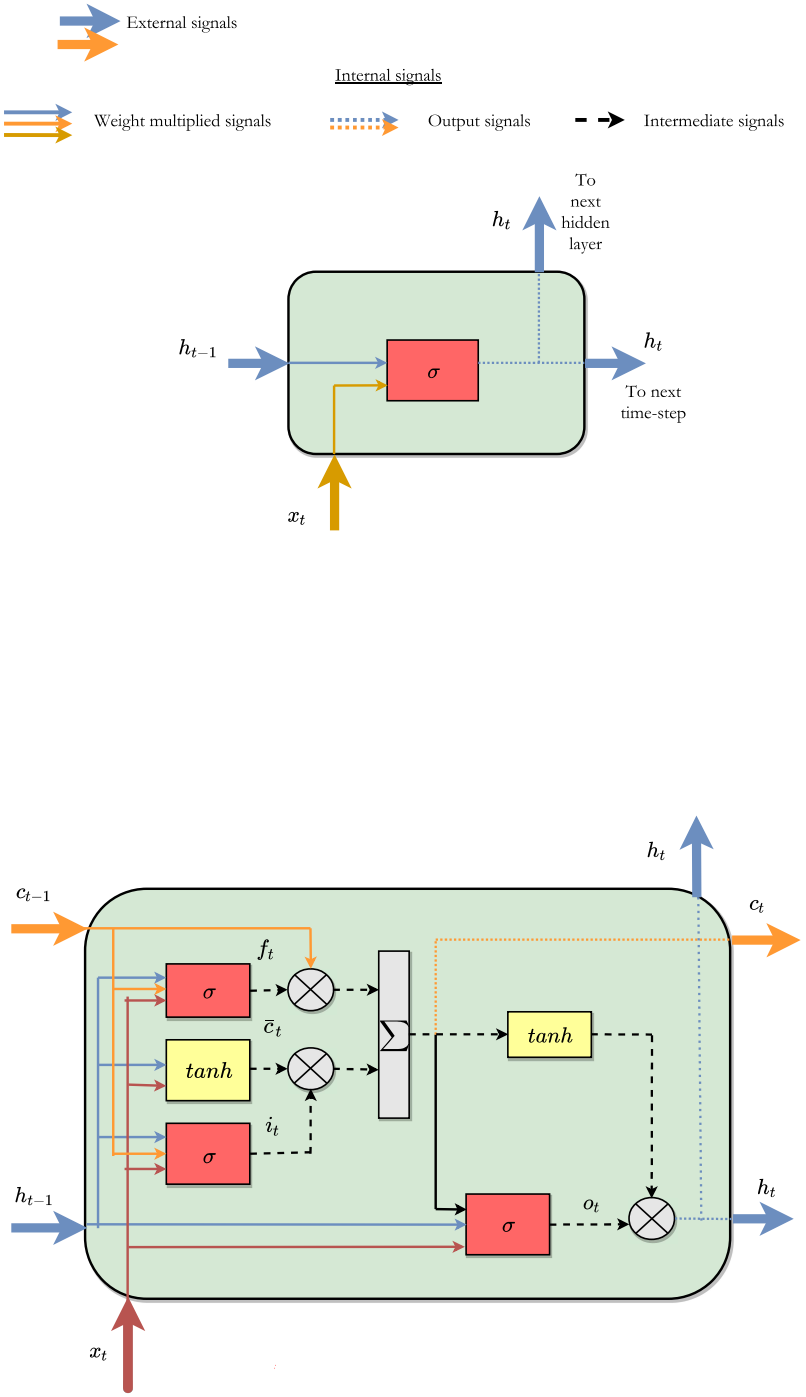


Figure 2.7 *An LSTM memory block(bottom) contrasted with a simple RNN block block(top).*

Bidirectional RNNs (BRNNs) were introduced in [36] and consist of two separate layers, one reading the same sequence in an opposite temporal order than the other, thus exploiting context from both past and future. A single layer of BRNN for eg., computes the following

$$\overrightarrow{\mathbf{h}_t} = f^h(\mathbf{W}^{\overrightarrow{xh}}\mathbf{x}_t + \mathbf{W}^{\overrightarrow{hh}}\overrightarrow{\mathbf{h}}_{t-1} + \mathbf{b}^{\overrightarrow{h}}), \tag{2.36}$$

$$\overleftarrow{\mathbf{h}_t} = f^h(\mathbf{W}^{\overleftarrow{xh}}\mathbf{x}_t + \mathbf{W}^{\overleftarrow{hh}}\overleftarrow{\mathbf{h}}_{t+1} + \mathbf{b}^{\overleftarrow{h}}), \tag{2.37}$$

where the notations are the same as uni-directional RNN only with added arrows to denote the direction. The above activations are summed before feeding to the next layer.

Bidirectional LSTMs (BLSTMs) are BRNNs with LSTM memory blocks. LSTMs and BLSTMs in combination with convolutional networks make up the current state-of-the-art for plethora of challenging tasks such as acoustic modeling in speech recognition [37], language modeling [38], machine translation [39] and image captioning [40].

2.6.4 Ensemble learning

It has been proved that learning of continuous-valued functions using neural network *ensembles* has a vast potential to improve the accuracy as well as leading to reliable estimates of generalization error [41]. Even combining the predictions of independently trained networks by simply averaging the outputs as

$$\mathbf{y}_{BEM} = \frac{1}{N_{ens}} \sum_{i=1}^{N_{ens}} \mathbf{y}_i, \tag{2.38}$$

where $\mathbf{y}_i$ are the individual network predictions, $\mathbf{y}_{BEM}$ is the output prediction from the ensemble, N_{ens} is the ensemble size, reduces the prediction MSE by a good amount. This method is often referred to as *basic ensemble method* (BEM) [42]. In particular, assuming that errors made by the individual networks are independent, it has been proved that

$$E(G, H_{BEM}) = \frac{1}{N_{ens}} \sum_{i=1}^{N_{ens}} E(G, H_i), \tag{2.39}$$

where H_i are the individual hypotheses and H_{BEM} is the ensemble hypothesis. This powerful result though holds true only for ensembles of smaller sizes, because, as

N_{ens} gets large, the original assumptions on the errors eventually break down. A more general case of *general ensemble method* (GEM) is discussed in [42].

2.7 Supervised speech separation

In a machine learning viewpoint, speech separation can be seen as "inferring" clean speech from noisy speech utterances. Under this view, the problem can be formulated as a supervised learning problem wherein features $\mathbf{x}$ are extracted from noisy speech and a model is trained to predict targets $\mathbf{y}$ corresponding to clean speech. Hence, during training, premixed signals (clean speech and background) need to be available to curate a dataset $\mathcal{D}$ of feature-target pairs $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), ..., (\mathbf{x}_N, \mathbf{y}_N)\}$. At the evaluation phase, predictions of appropriate targets are obtained as outputs from the trained model. Given sufficient training data, supervised speech separation methods have been proved to very successful and is also a preferred choice for real-time implementation due to its frame-wise operation [13].

The major challenges in these methods are in choosing appropriate features that can discriminate target speech from interference, the choice of classifier/regressor and the choice of target. The multi-channel methods differ from the single-channel methods only by the way features are fed into the framework and is discussed in the following sub-section.

2.7.1 Features

Spectral audio Features

The STFT-based log magnitude spectrograms and log mel spectrograms are the simplest amplitude based features and have been successfully used in speech separation respectively in [43] [44, 45]. The other highly successful feature used in numerous audio based pattern recognition and classification systems is *Mel frequency cepstral coefficients* (MFCCs). They are computed by taking discrete cosine transform (DCT) of log-magnitude values in mel scale spectrogram (refer to Figure 2.8. DCT aids in obtaining smooth spectral shape corresponding to vocal tract by discarding other components due to glottal variation. MFCCs were one among the features used for supervised speech separation in [13]

Features such as *linear prediction coefficients* (LPCs) are useful in obtaining speech specific information for learning. LPCs are filter coefficients of an all-pole filter which model the speech production process — for a signal downsampled to 8 kHz, lower orders such as 12 (the general rule of thumb for order choice being

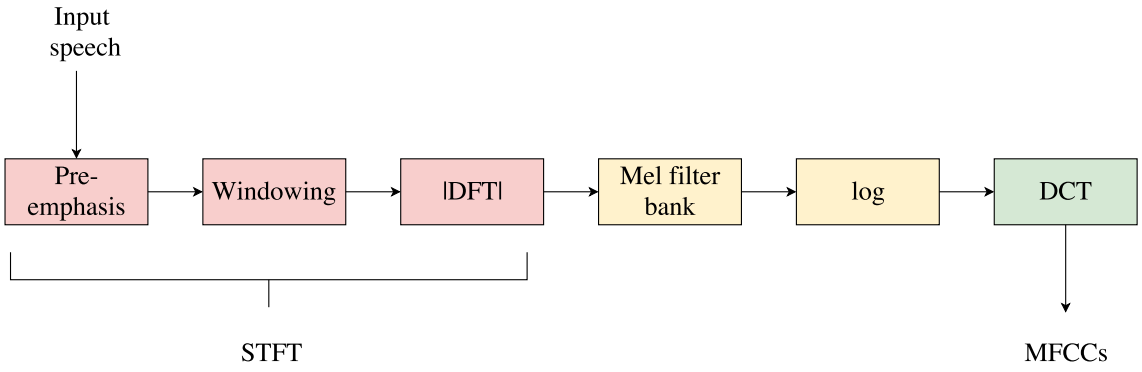


Figure 2.8 *Steps in the extraction of MFCC*

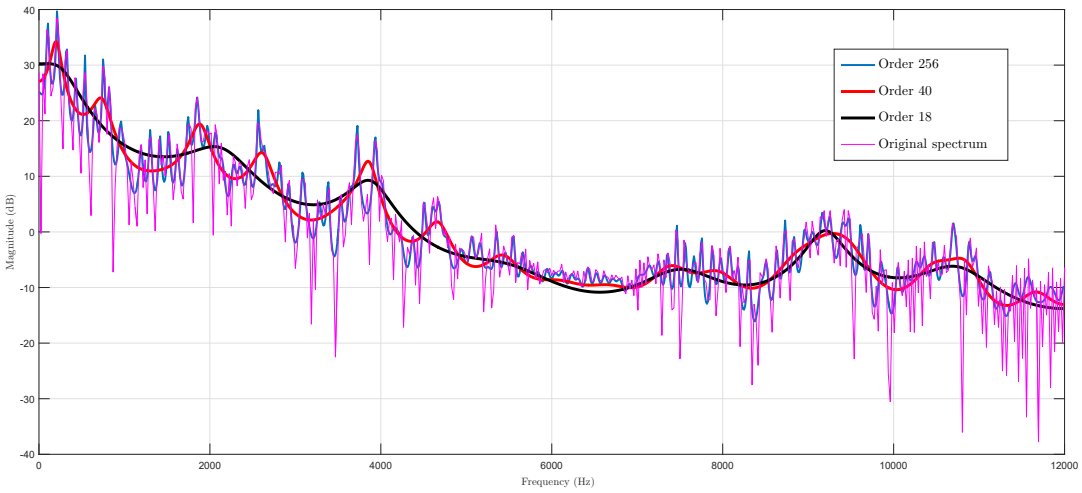


Figure 2.9 *LPC spectra illustrated. LPC of order 18 captures well the vocal tract shape for a spoken vowel signal sampled at 24 kHz. Extraction of LPCs over short time frames thus gives speech-related cues.*

$4 + f_s/1000$, f_s is the sampling frequency in Hz), capture the broad spectral peaks corresponding to the vocal tract resonances, while higher orders such as 40 also captures the glottal variations (refer to Figure 2.9.

Apart from the above commonly used features, other sophisticated features such as *gammatone filter-bank power spectra* [13], *gammatone frequency cepstral coefficients* (GFCCs) , *relative spectral transform with perceptual linear prediction* (RASTA-PLP) [13], *amplitude modulation spectrogram* (AMS) [46, 47, 13] — often used in combination—have been particularly used for speech separation/ enhancement by learning masks.

Spatial audio features

Spatial features come in the fore-front when using multiple microphones. While arguably the spectral features from multiple microphones can be concatenated to form an extended feature vector, they are likely to overfit to the specific microphone placement/ characteristics seen in training [48]. Moreover, naturally it leads to large feature dimension with high correlation among features — not a desirable quality when using neural networks. *Inter-channel phase difference* (IPD) and *inter-channel level difference* (ILD) are the two traditionally used features for sound source localization based on a stereo or binaural recording with two microphones, akin to human ears. For M microphones and for every microphone pair (i, j), the frequency dependent IPD and ILD(dB) can be extended as

$$\mathrm{IPD}_{i,j}(n,k) = \angle m_i(n,k) - \angle m_j(n,k), \tag{2.40}$$

$$\mathrm{ILD}_{i,j}(n,k) = 20\log_{10}\left(|m_i(n,k)|\right) - 20\log_{10}\left(|m_j(n,k)|\right). \tag{2.41}$$

But the increase in number of microphones M, hence also the number of unique microphone pairs $(= M(M-1)/2)$, leads to a quadratic increase in the spatial vector dimension. In order to avoid this, a phase based feature vector is proposed in [49] and is derived in the Section 3.2 .

Other features

It has also been conjectured that higher level information from the language model of an ASR system can in turn aid in better separation [7, 48](a language model is an integral component of an ASR system. A brief description of ASR is given in the next section). The integration of language models to speech separation is not new and has already been attempted in model-based speech separation approaches such as [50].

2.7.2 Training targets

Supervised learning has been applied to speech separation in many different forms, but we focus on those methods that are based on TF representations, which can be categorized on the basis of targets as (i) *mapping* based targets and (ii) *masking* based targets (refer to Figure 2.10 [13].

Mapping based targets are the spectrograms — real or complex-valued — of

clean speech and hence the learning problem can be formulated as a multi-variate regression problem. The objective function for such a problem can thus be formulated as

$$E_{\text{map}}(\hat{s}) = \sum_{k,t} D\big(\hat{s}(n,k), s(n,k)\big), \tag{2.42}$$

where D is a distance measure, typically squared Euclidean distance, $\hat{s}(n,k)$ and $s(n,k)$ are the estimated and true source signal spectrograms respectively.

On the other hand, masking targets are *TF masks* or *TF filters* — binary-valued, real-valued or complex-valued — which are estimated and multiplied with the mixture spectrogram to obtain the clean signal. Masking based approaches can be of both regression type and classification type depending on the type of mask used and the objective function used. If the objective is to estimate the mask which best approximates the target mask, the approach can be called *mask approximation approach*[51]. The objective function for such an approach is then

$$E^{MA}_{\text{mask}}(\hat{y}) = \sum_{k,t} D\big(\hat{y}(n,k), y(n,k)\big), \tag{2.43}$$

where $\hat{y}(n,k)$ and $\hat{y}(n,k)$ are the estimated and ground truth masks respectively, E^{MA}_{mask} is the objective function.

With masking based targets, the optimization can also fall under *signal- approximation* type [51] wherein an estimate of clean spectrogram is first obtained by applying the TF mask over the mixture spectrogram as $\hat{s}(n,k) = |m(n,k)|\ \hat{y}(n,k)$ and then using Equation 2.42 for optimization.

For obvious reasons, estimating IBM is thus a binary classification problem and has been a widely used target in supervised separation [46, 52, 53, 54]. Due to its continuous valued nature, estimating IRM is a regression problem. For the same reason, it is also less susceptible to musical artifacts.

There are other possible ratio masks such as *amplitude ratio mask*

$$y_R(n,k) = \frac{|s(n,k)|}{|m(n,k)|}, \tag{2.44}$$

which does not require noise estimation, uses the mixture signal $m(n,k)$ directly and can take values in the range $(0,\infty)$; and its extension called *complex ratio mask* [55]

$$y_{CR}(n,k) = \frac{s(n,k)}{m(n,k)}, \tag{2.45}$$

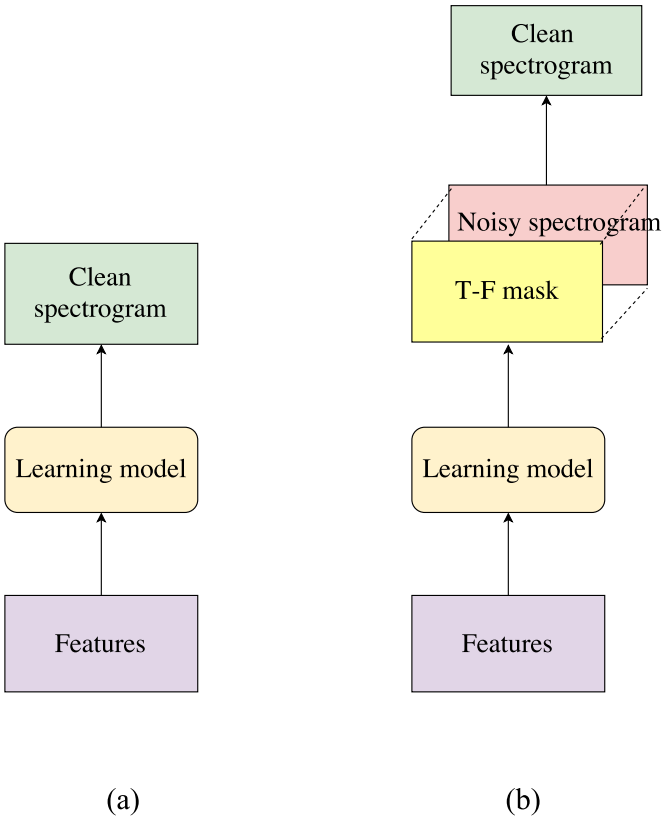


Figure 2.10 *Two types of TF representation based supervised speech separation — (a) mapping of features to clean spectrogram, (b) mapping of features to a TF mask, which is then applied over the noisy spectrogram to get the clean spectrogram.*

to include phase information. The real and imaginary parts in a complex ratio mask can be jointly learned, with a pit-fall that the values can now be in the range $(-\infty, \infty)$ leading to potential problems in learning. Typically the range of mask values is compressed to a finite range by clipping or soft truncation [55].

2.8 Noise-robust automatic speech recognition

A bird's-eye view of a typical robust ASR system, shown in Figure 2.11, is given in this subsection. While MFCCs and PLPs are the fundamental features used in ASR systems, for increased robustness to speaker and environment variations one or more feature transformations are applied such as *cepstral mean subtraction (CMS)* — to remove linear channel distortions — , *cepstral mean and variance normalization* (CMVN) — to increase robustness and reduce mismatch between training and test conditions —, *relative spectral transformation* (RASTA) — to further clean up PLP by getting rid of non-linguistic information.

The inherent variability in the acoustic signal due to background noise mostly manifests as a *mismatch* in training and testing conditions — the mismatches can also be in terms of reverberation time, direct-to-reverberent ratio, SNR conditions, noise characteristics. In a multi-microphone setting, it can also be due to the number of microphones, their spatial locations and their frequency responses. The mismatch problem due to background noise have been addressed broadly by 2 classes of methods: *feature-based* and *model-based* [44].

The feature-based methods may include using robust features — RASTA [56], advanced front-end (AFE) [57] features — or using an enhancement algorithm that transforms the noisy features to closely match the distribution of training data [58]. The use of feature-based methods thus results in having a speech separation/ enhancement front-end and the methods described in this thesis will fall under this class. [44] further classifies feature-based methods into two groups depending on the availability of stereo training data — stereo implying the availability of noisy and the corresponding clean signals. Supervised speech separation methods — such as predicting T-F masks to perform feature enhancement — obviously needs this stereo data. Methods that do not have access to this stereo data use prior knowledge about speech and/or noise to perform feature enhancement.

In model-based approaches on the other hand, the acoustic model parameters of the ASR are themselves adapted — this way the distribution of noisy or enhanced features are matched with that of training data [59, 58]. This is the functionality of the *adaptation* block in the Figure 2.11. The remaining blocks, common to all ASR systems, will be shortly described next.

Given a sequence of observed features, the decoding step finds the optimal sequence of words utilizing additional knowledge in the form of acoustic and language models. The speech recognition problem has a Bayesian formulation as

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \; P(\mathbf{W}|\mathbf{O}), \tag{2.46}$$

$$= \arg\max_{\mathbf{W}} \; \frac{P(\mathbf{O}|\mathbf{W}) \; P(\mathbf{W})}{P(\mathbf{O})}, \tag{2.47}$$

$$= \arg\max_{\mathbf{W}} \; P(\mathbf{O}|\mathbf{W}) \; P(\mathbf{W}), \tag{2.48}$$

where $\mathbf{O}$ is the observed feature sequence, $\mathbf{W}$ is the observed word sequence and $\hat{\mathbf{W}}$ is the optimal word sequence. In small vocabulary speech recognition, $P(\mathbf{O}|\mathbf{W})$ is computed for each of the words, whereas for larger cases, words are modeled as a

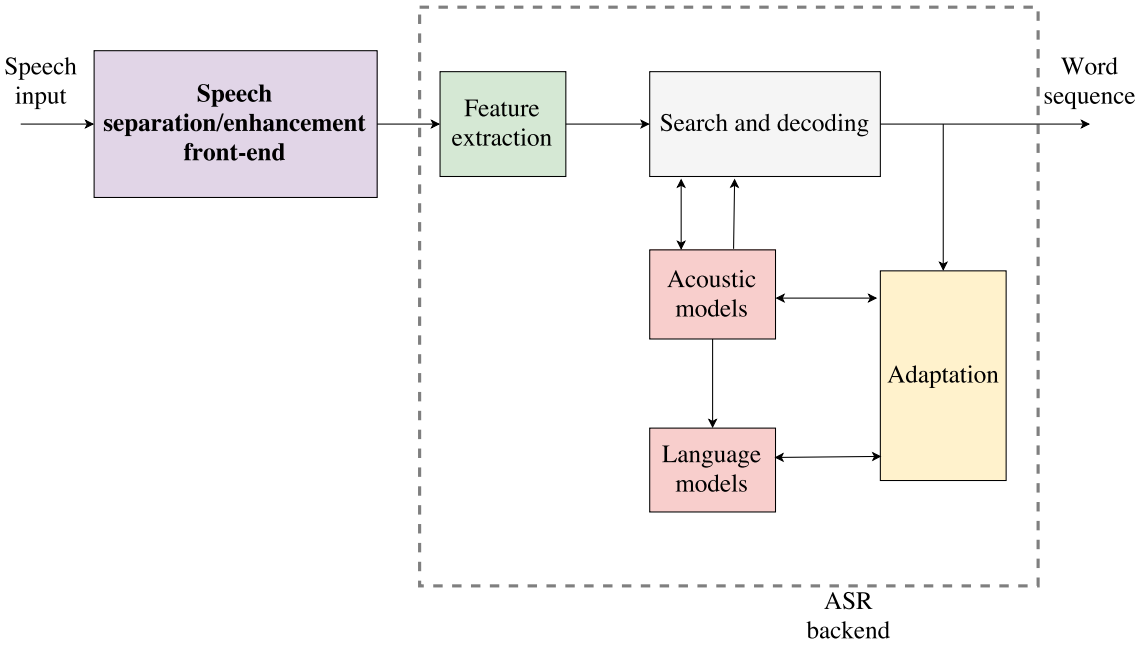


Figure 2.11 *A typical robust ASR framework. The main focus of this thesis will be on speech separation/ enhancement front-end. The back-end will be fixed and used only for assessing the usefulness of proposed front-end.*

sequence of phonemes represented as a sequence of states $\mathbf{S}$ as

$$P(\mathbf{O}|\mathbf{W}) = P(\mathbf{O}|\mathbf{S})\ P(\mathbf{S}|\mathbf{W}), \tag{2.49}$$

$P(\mathbf{O}|\mathbf{S})$ is represented by the *acoustic model*, $P(\mathbf{S}|\mathbf{W})$ by the *pronunciation model* and $P(\mathbf{W})$ by the *language model* respectively. Traditional acoustic models used *Gaussian mixture model based hidden Markov models* (GMM-HMM). With the success of deep learning, DNNs have replaced GMM-HMMs — the current state-of-the-art ASR employs CNN-RNN based acoustic models [37]. The *n-gram* language model, most commonly a *trigram* model, acts as a prior and helps in guiding and constraining the search among the alternative word hypothesis. The language model is also witnessing the shift to RNNs [60].

Another interesting recent observation is that deep learning results in ASR have shown preference for primitive features such as log-mel filterbank outputs, rather than MFCCs. In multi-microphone case, using direct time-domain signals have shown tremendous promise in learning acoustic models [61].

In a *speaker independent* (SI) speech recognition, typically a single acoustic model is trained over as much data as possible, covering variability over gender, ac-

cent and other speaker-specific idiosyncrasies. *Speaker adaptation* techniques strive to bring the performance of SI systems close to *speaker dependent* systems — the latter can yield 2-3 times lower word error rate (WER — an evaluation metric defined briefly in section 3.4, provided they are trained with the same amount of data. Speaker adaptation in GMM-HMMs is achieved by techniques like *maximum likelihood linear transformation* (MLLT), *feature-space maximum likelihood regression* (fMLLR), *speaker adaptive training* etc. Speaker adaptation in DNN based acoustic models have been an active research area and solutions that have been proposed include using *i-vectors* along with raw acoustic frames; appending additional layers to the model etc. [62].

2.9 Standard databases, evaluation challenge competitions and the CHiME-3 challenge

The progress in the field of ASR has been largely attributed to the regular publication of standard datasets and worldwide evaluation campaigns conducted around it — these massively contributed towards fair comparison of the diverse methods proposed worldwide. Identifying and designing evaluation challenge tasks is of utmost importance [63]: tasks should be compact so that efficient evaluation is possible; should be realistic that 'toy' solutions are not accepted; should address an unsolved or untested problem in the area; should make it impossible for the developers to tune their systems to the test corpus; last but not the least, the forthcoming challenges should identify and address the issues and problems of preceding challenges. The history of such evaluation races dates back to the series of competitions launched as part of the DARPA Resource Management project in 1987.

The main focus in the last twenty years has been the evaluation of noise-robustness in ASR systems, kick-started by the Aurora 2 and Aurora 4 corpora [63]. Early evaluations (in early 2000s) in this area had difficulties in capturing the acoustics of real applications — for example, Aurora 2 and 4 did not capture the channel variability of real acoustic mixtures as they provided only instantaneous mixtures of speech and noise, apart from other problems such as using only short segments of noise samples and providing no opportunity to model noise context. Later evaluations were targeted at capturing the real-world acoustics such as that of meeting rooms and lecture halls where high SNRs generally prevail but in distant microphone scenarios and employing microphone arrays.

On similar lines, the first challenge of the CHiME series — PASCAL CHiME speech separation and recognition challenge (CHiME-1) [64] — dealt with keyword

identification from highly reverberant, domestic audio recordings involving highly dynamic and complex acoustic backgrounds, recorded using binaural microphones in a distant-talking scenario. The CHiME-2 challenge [65] was developed on top of CHiME-1, focusing on larger vocabulary and considering the talker movements as well. Both the challenges used artificially mixed speech and noise; while in the first challenge the clean utterances were convolved with fixed binaural room impulse response, the second challenge used a time-varying response by simulating talker movements.

The CHiME-3 challenge [66] targeted mobile speech recognition using a multi-channel mobile tablet and pushed the level of realism one step further by capturing speech data directly in diverse noisy settings — in 4 different environments — containing multiple unknown sound sources. Thus difficulties spanned across a continuum of SNR variations and as well as degrees of reverberation. In addition to real data, it also included a simulated training data constructed by using time-varying impulse responses in such a manner that the microphone responses, SNRs and effects of speaker movements are matched with the real recordings. The details are in [1] and [63]

2.9.1 Dataset mismatches in the context of noise-robust ASR

In most of the existing corpora, speakers forming the training data are different from the test data and this is often considered a good practice; such mismatch in speakers can be addressed by speaker adaptation methods. But the acoustic conditions — such as reverberation time (RT60), SNR, direct-to-reverberant ratio (DRR) or noise characteristics [67] — of the test data usually match with that of training data, thus methods fail to generalize in mismatched acoustic conditions. With regard to multi-microphone methods, the number of microphones, their spatial positions, their geometry, their frequency responses, should also be taken into account [67].

Some issues are of particular concern with methods based on neural networks. Traditional speech enhancement techniques did not encounter an issue due to mismatched noise conditions, but this becomes a major concern with these methods as they are bound to be useful only when trained with diverse training material [67]. Since most of the corpora rely on the simulated data for increasing the amount of data — again deemed incredibly useful for training the neural networks —, the mismatch between them and the actual real data becomes a new issue and addressing them is of utmost importance. [67] particularly mentions about the suspicion prevailing in speech processing community regarding simulated data because of the

misleadingly high performance of DoA-based adaptive beamformers on simulated data compared to real data.

CHiME-3 challenge was designed to revolve around in studying all these mismatches and two very recent papers by the challenge organizers themselves — [63] and [67] — analyse these aspects exhaustively.

3. METHODS

Figure 3.1 shows the overall supervised speech separation framework; the individual methods are explained in this chapter.

3.1 TDoA estimation, beamforming and data processing

The recording device is a commercial tablet phone additionally fitted with with 6 microphones — 5 forward facing to capturing the target speech and one backward facing capturing the background noise. The array input signals are first RMS normalized so as to reduce gain mismatches.

In the TDoA estimation, it is assumed that the position of the speaker holding the tablet is static — that he/she does not move significantly throughout the utterance. This way, the microphone array signals can be time-aligned only once for every utterance using the TDoA values estimated at sub-sample resolution — resolution less than the sampling time duration $1/f_s$. GCC-PHAT method described in section 2.5.1 with *parabolic peak interpolation* [6] is used in estimating TDoAs (refer to Figure 3.2). The microphone index $j = 0$ is arbitrarily chosen as the reference microphone in the computation of relative time delays. The computed TDoA values are then used to steer the DSB beamformer.

In obtaining the TF representations, the array input, the DSB output and the close-talk microphone signals are processed in 32 ms overlapping frames with a 50% overlap using square-root Hann window. STFT is computed using 512-point DFT (effectively 257 frequency bins) and down-scaled to 30 mel bands. The predicted TF mask (in mel frequency scale) is then converted back to STFT resolution before applying over the magnitude spectrogram of the DSB output.

3.2 Computation of features and targets

As spectral features, MFCCs are extracted from the DSB output and the signal from backward facing microphone; LPCs are also extracted from the DSB output after pre-emphasis and concatenated. For incorporating spatial cues, a phase-based feature proposed in [49, 68] is used. This feature circumvents the problem of increase

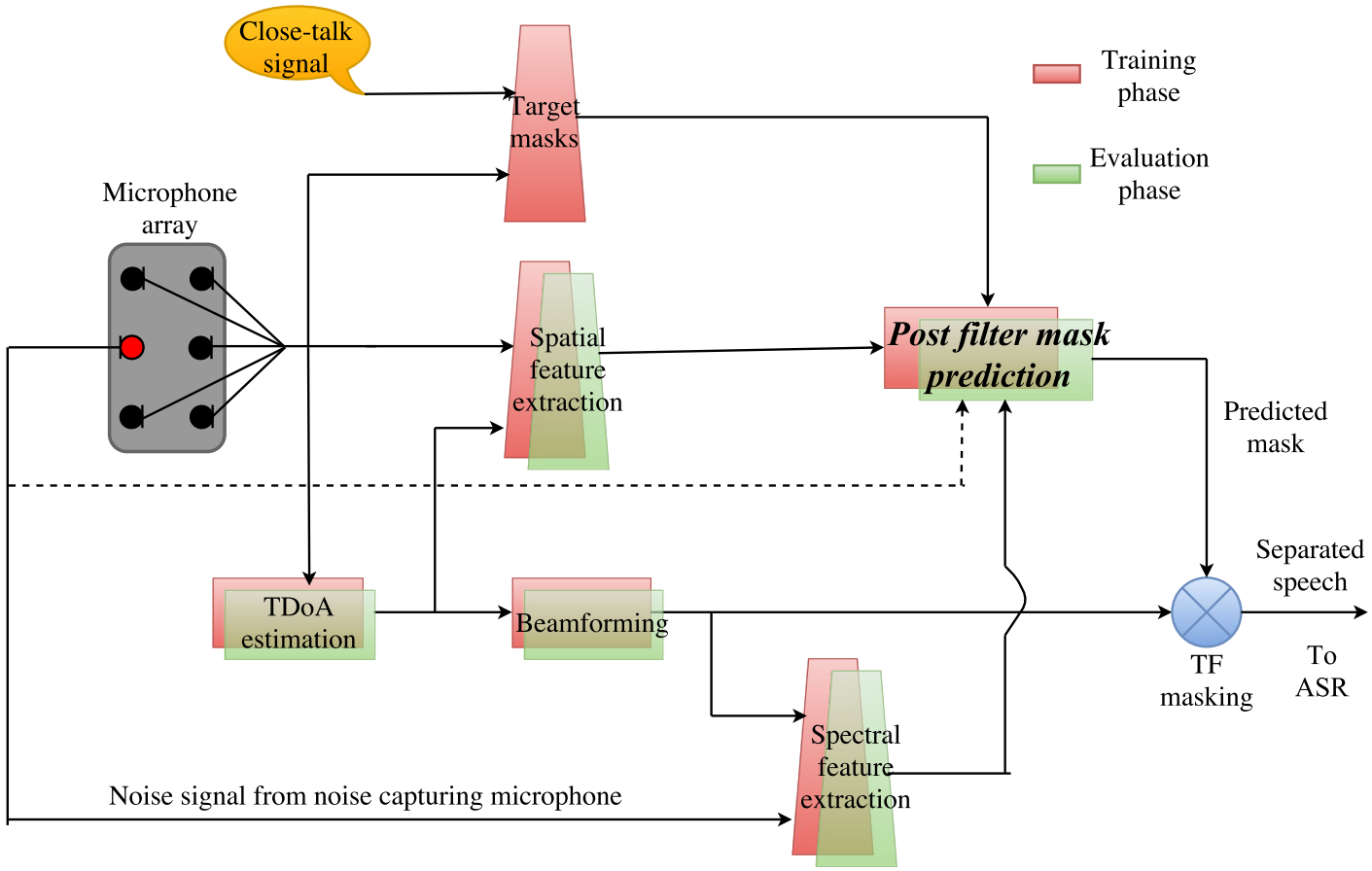


Figure 3.1 *The big picture of methods used. The recording device is a commercial tablet phone additionally fitted with 6 microphones. TDoAs estimated by GCC-PHAT steer the DSB beamformer. The post-filter consists of a BLSTM-RNN model trained to predict TF masks based on spectral and spatial features. During training, target masks are computed using the clean speech from a close-talk microphone.*

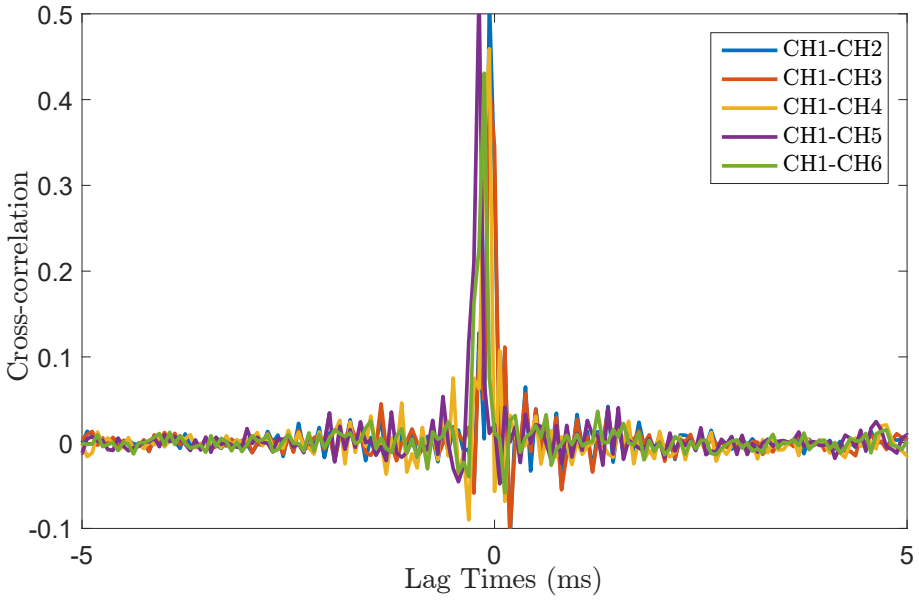


Figure 3.2 *Illustration of cross-correlation between microphone signal pairs obtained by GCC-PHAT for an utterance from DT set of CHiME-3 data.*

in the feature vector dimension if the IPD and ICD — defined in section 2.7.1 — are directly used for every microphone pair. The computation of this feature is explained as follows.

Using the propagation model described in section 2.5.1, in an ideal noise and interference-free conditions, the phase component of jth microphone signal can be written as

$$\phi_j(n,k) = \phi_s(n,k) - \omega_k \tau_d^{(j)} \tag{3.1}$$

where $\phi_s(n,k)$ is the phase of the source signal, $\tau_d^{(j)}$ is the propagation delay and $\omega_k = \frac{2\pi k}{N_{\text{DFT}}}$ is the angular frequency. From this definition and from the definition of TDoA, TDoA can be written as

$$\Delta\tau_d^{(i,j)} = \tau_d^{(j)} - \tau_d^{(i)} = \omega_k^{-1}\big(\phi_j(n,k) - \phi_i(n,k)\big) \tag{3.2}$$

On rearrangement, we get

$$\phi_j(n,k) - \phi_i(n,k) = \omega_k \Delta\tau_d^{(i,j)} \tag{3.3}$$

In presence of competing sources — whose signals come from a direction other than the target source direction — the measured instantaneous difference in phase $\big(\angle m_j(n,k) - \angle m_i(n,k)\big)$ will not agree with the phase difference — computed over

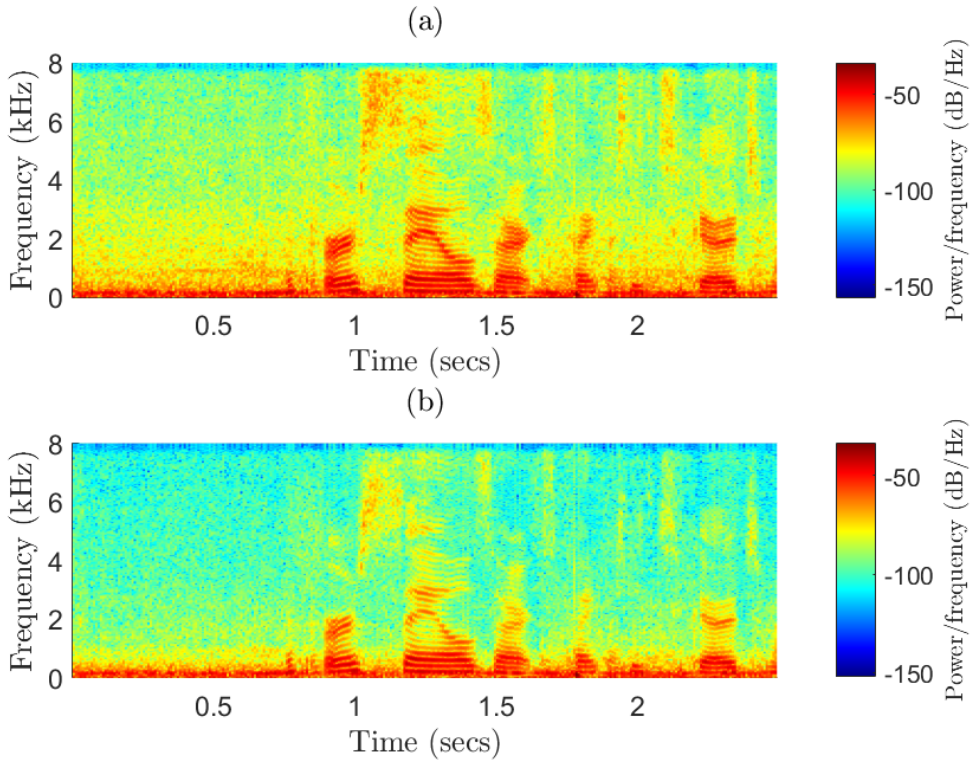


Figure 3.3 *(a): Mixture signal from one of the microphones for an utterance from DT set of CHiME-3 data; (b): Corresponding DSB output. Applying a post-filter to the DSB output further cleans up the noisy utterance.*

an entire utterance — as defined in equation 3.3. This disagreement is quantified for every microphone pair $\{i, j\}$ and an aggregate feature is computed as

$$\psi(n,k) = \frac{2}{J(J-1)} \sum_{\forall\{i,j\}} \cos\big((\angle m_j(n,k) - \angle m_i(n,k)) - \omega_k \Delta\tau_d^{(i,j)}\big) \tag{3.4}$$

The above feature thus have values in the range $[-1, +1]$ — a value of $+1$ indicating perfect agreement at a particular TF point and vice-versa.

Two different TF masks will be used and compared — Wiener filter and log-ratio mask. Wiener filter is defined as

$$y_{Wiener}(n,k) = \frac{|s(n,k)|^2}{|s(n,k)|^2 \ + \ |v(n,k)|^2} \tag{3.5}$$

where the source signal spectrogram $s(n,k)$ can be obtained from the close-talk microphone. The noise spectrogram need to be estimated and is done in 2 steps

[1]. The first step involves estimating the STFT domain impulse response between the close-talk microphone and every tablet microphone. Subsequently, the signal from close-talk microphone is convolved with the estimated impulse responses and subtracted from every microphone signal to get the noise estimates at every microphone. The individual Wiener filters are then averaged to yield the final target Wiener filter mask.

The log-ratio mask is defined as

$$y_{LR}(n,k) = 20\log_{10}\left(\frac{|s(n,k)|}{|m(n,k)|}\right) \tag{3.6}$$

where $|m(n,k)|$ is the average of the magnitude spectra of individual microphone signals. It has to be noted that while Wiener filter mask values are bounded in the range $[0,1]$, log-ratio mask values can go unbounded in $\mathbb{R}$ and may need to be truncated appropriately.

Finally, in order to speed up the training by neural networks, all the feature vectors are shifted and scaled to have zero mean and unit standard deviation. For this, all feature vectors from all the utterances constituting the training data are concatenated. For the qth feature index of pth feature vector the normalization is done as

$$\tilde{x}_q^{(p)} = \frac{x_q^{(p)} - \mu_q}{\sigma_q} \tag{3.7}$$

where μ_q and σ_q are the mean and standard deviation respectively of the qth feature. A similar shifting and scaling is also done to log-ratio mask vectors.

3.3 Proposed post-filter training and mask prediction

The TF mask prediction can be seen as a sequential data prediction problem for which RNNs ought to be the right choice, and has already been applied successfully to single channel separation/ enhancement problems. Though extensions to multi-channel cases have been proposed sparsely, the idea of using it as a post-filter to beamformer has only been studied in [49, 68] which uses an FFNN with multiple hidden layers to map spatial feature vectors to Wiener filter mask vectors. By replacing FFNN with a BLSTM RNN, we hope to overcome the limitation of dealing with only short context.

Formally stated, given a sequence of feature vectors $\mathbf{X} = \{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^T\}$, $\mathbf{x}^t \in \mathbb{R}^{N_f}$, where N_f is the feature vector dimension, T is the sequence length (in number of frames) of the BLSTM model, the mask prediction problem can be stated

as estimating a sequence of mask vectors $\hat{\mathbf{Y}} = \{\hat{\mathbf{y}}^1, \hat{\mathbf{y}}^2, ..., \hat{\mathbf{y}}^T\}$ that minimizes the mean squared error (MSE) between the target $\mathbf{Y} = \{\mathbf{y}^1, \mathbf{y}^2, ..., \mathbf{y}^T\}$ and $\hat{\mathbf{Y}}$, where $\mathbf{y}^t$ and $\hat{\mathbf{y}}^t \in \mathbb{R}^{N_t}$, N_t is the dimension of the output TF mask vector — depends on DFT size N_{DFT} if linear resolution is used or the number of mel bands N_b if downscaled to mel-scale resolution (refer to Figure 3.4. This method — in the supervised speech separation context — is thus a mask-approximation approach.

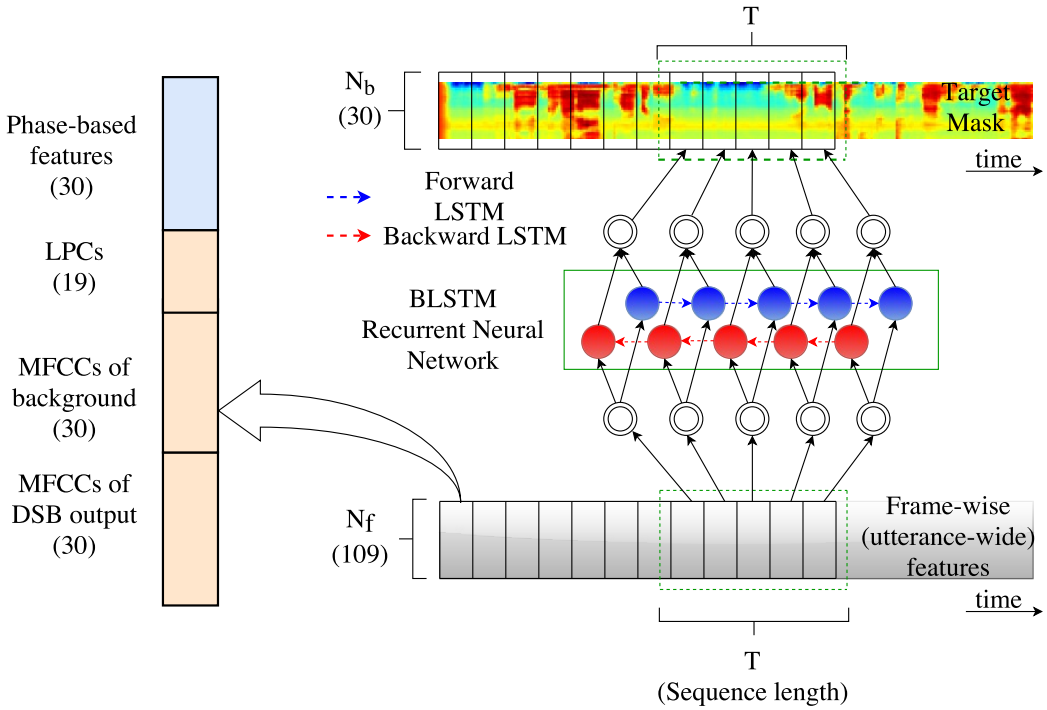


Figure 3.4 *Illustration of sequence-to-sequence learning using BLSTMs. For a feature vector sequence of length T time frames, the BLSTM-based model should learn to predict the target TF mask vector sequence of the same length T.*

It is not possible to know the best configuration of the network a priori and hence several networks have to be trained with different architectures — by varying the number of hidden layers and number of LSTM cells in each layer. The input layer size will be the same as the feature dimension, while the output layer size depends on the resolution of the TF mask used. Because the choices of the TF mask have been fixed — Wiener and log-ratio, both of them being soft masks —, we are solving a regression problem and the popular choice of linear output activation will be used. LSTM units typically use *tanh* or *softsign* activation functions; the choice between them will be left to the hyperparameter search. The choice of sequence length T for the sequences will also be a parameter in hyperparameter search.

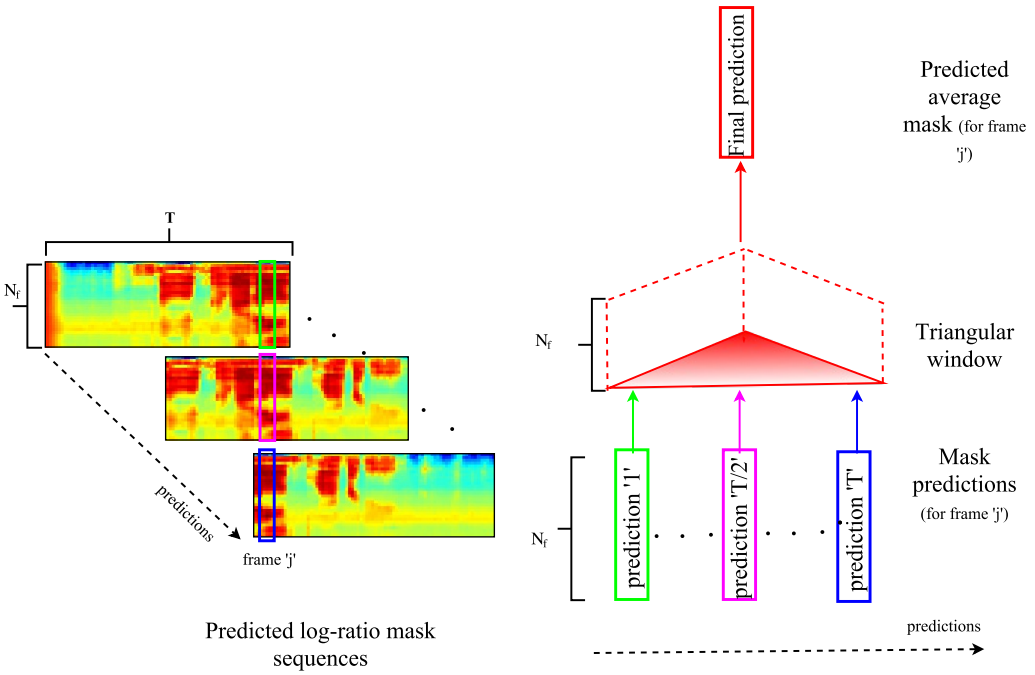


Figure 3.5 *Illustration of mask prediction for test data.*

For training, Adam [29] — a stochastic gradient-based optimizer — is used to learn the parameters of the model. *Early-stopping* will be the primary regularization technique used; a *patience* parameter will be specified in terms of number of epochs — if the validation MSE does not improve for the number of epochs specified as patience, the training is halted. There are number of ways to choose the final weights; here, the weights corresponding to the epoch where the improvement had just started to cease will be chosen. Finally an ensemble of 5 networks with different random weight initialization are trained.

When using the trained models for predicting the masks for test data, overlapping feature sequences of length T are formed with a single frame-hop. This results in T predictions for a single frame, which are then combined by a triangular window as illustrated in Figure 3.5. Further smooth predictions are obtained by simply averaging the outputs of individual networks forming the ensemble. The overall steps in the machine learning pipeline is illustrated in Figure 3.6

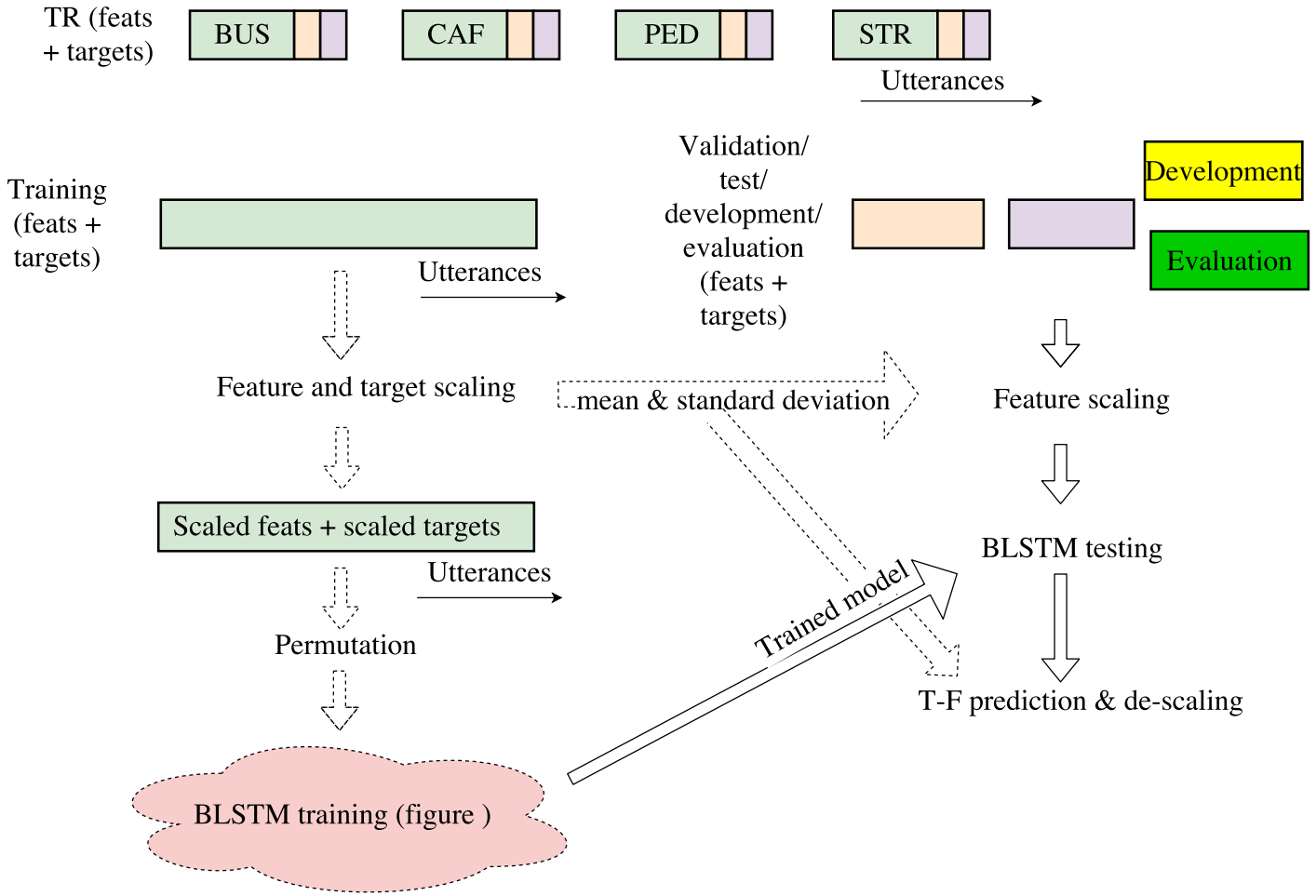


Figure 3.6 *Steps in the machine learning pipeline for TF mask prediction using the CHiME-3 data. The dataset is explained in chapter 4.*

3.4 Evaluation measures

The mask prediction procedure described before results in TF masks which are close to the target mask in an MSE sense. But the ultimate goal of speech separation is to improve speech quality and intelligibility which are best assessed by conducting listening tests, but they are expensive and time consuming. Instead relevant objective measures can be picked from several instrumental speech-quality and speech-intelligibility measures. The two objective measures which will be used in the evaluation of our speech separation framework are described next, followed by the word error rate measure used in evaluating the ASR back-end.

3.4.1 Short term objective speech intelligibility (STOI)

STOI [69] is an objective intelligibility measure which is shown to be highly correlated with human speech intelligibility. Compared to other speech intelligibility measures which typically rely on global statistics across entire sentences, STOI is a correlation-based measure based on short-time segments. In particular, STOI has been shown to be more successful — compared to other objective intelligibility measures which are derivatives of articulation index (AI), speech intelligibility index (SII) and speech transmission index (STI) [69] — when evaluating TF weighted noisy speech or other noise-reduced speech with added non-linear distortion. The STOI algorithm yields a scalar output in the range $[0, 1]$, which is shown to exhibit a monotonic relationship with the average intelligibility measured in terms of percentage of correctly understood words, averaged across a group of people [69].

3.4.2 Frequency-weighted segmental SNR (fwSNR)

fwSNR [70] is another objective measure highly correlated with speech intelligibility and is defined as

$$\text{fwSNR} = \frac{10}{N_{\text{frames}}} \sum_{n=0}^{N_{frames}-1} \frac{\sum_{k=1}^{K} w_{fw}(n,k)\ log_{10}\ \frac{|s(n,k)|^2}{|s(n,k)-\hat{s}(n,k)|^2}}{\sum_{k=1}^{K} w_{fw}(n,k)} \tag{3.8}$$

where the frequency bands k are proportional to ear's critical bands; the weights $w_{fw}(n,k)$ are based on the magnitude spectrum of clean signal raised to a power of 0.2 [70]; $s(n,k)$; $\hat{s}(n,k)$ are the critical band spectra of close-talk and separated speech signals respectively; N_{frames} and K are the total number of time frames and frequency bands respectively. In this work, 25 mel bands are used and SNR values

are limited between $[-10, 35]$ dB [70].

3.4.3 Word error rate (WER)

WER is a common metric used to assess the performance of ASR systems and requires a transcription in the form of correct word string for its computation. It is defined as

$$\text{WER} = \frac{S + D + I}{N_r} \tag{3.9}$$

where S, D, I are number of substitutions, deletions and insertions respectively, and N_r is the total number of words in the reference transcription. Insertion refers to the additional words introduced, deletion to the missing words and substitution to the replacement of words respectively by the ASR system compared to the reference sequence.

4. EVALUATION

4.1 CHiME3 challenge data

The 3rd CHiME Challenge [1] data consists of spoken sentences from a subset of Wall Street Journal (WSJ) corpus comprising of 5000-word vocabulary. The signals were captured using a tablet equipped with 6 microphones — 5 forward facing and one backward facing — in 4 different real-world noisy environments namely, public transport (BUS), pedestrian area (PED), cafeteria (CAF), and street (STR). The provision of backward-facing microphone aids in capturing noise effectively than in the far-field case, although some amount of speech leaks into it due to reverberation and diffraction of sound waves. The signals were sampled at 48 kHz, down-sampled to 16 kHz and quantized at 16-bit depth.

The dataset is divided into training (TR), development (DT), and evaluation (ET) sets, each containing recordings from all environments in both real and simulated settings. In real settings, there are 4 different speakers — two male and two female — for each set, totaling to 12 different speakers. The speakers were also equipped with a close-talk (lapel) microphone for acquiring clean signals. The talker-tablet distance, though varying, was maintained at around 40 cm—hence this is neither a close-talking scenario nor a far-field scenario.

Simulated mixtures were generated as follows: a block-wise least squares filter [1] was used to estimate the impulse response between close-talk microphone and the tablet microphones; the speech signals from the individual microphones were then removed and the resulting signals were then added with the booth recordings. Simulated mixtures for training data were obtained by mixing clean speech recordings from 84 speakers (recorded in an acoustically isolated booth) with separately recorded noise backgrounds. The specifics of the methods are detailed in [1]. There are totally 1600 real and 7138 simulated noisy utterances in TR set, 1640 and 1640 respectively in DT set, and finally 1320 and 1320 respectively in ET set.

4.2 Data curation for post-filter training

The features and target TF masks are extracted for TR, DT and ET datasets as described in Section 3.2. Speech source signals — required for creating target TF masks — are obtained from close-talk microphone for the real cases and the booth data for simulated cases respectively.

As per the challenge rules, only the data from TR set is used for post-filter training. 90% of data from TR set — all environments equally balanced — is used for actual training; the resultant set will be called TR-TR. For cross-validation, 5% of TR data is used and called TR-VAL; the remaining will be used in assessing the generalization error within the TR set and called TR-TEST.

To perform feature and/or target scaling, mean and standard deviations are computed from TR-TR data as described in figure 3.6. All other data: TR-VAL, TR-TEST, DT and ET will be first scaled using these values. To feed in the data to the Keras [71] deep learning library, features and targets from every utterance are tensorized as follows:

Table 4.1 *Data tensorization needed for Keras*

	Original shape	New shape
Features	(N_{frames}, N_f)	(N_{seq}, T, N_f)
Targets	(N_{frames}, N_b)	(N_{seq}, T, N_b)

In the above table N_{seq} is the total number of sequences of length T that can be carved out of an utterance. For TR set, the sequences are non-overlapping while for DT and ET sets, the sequences overlap with a single frame hop as illustrated in figure 3.5.

4.3 Neural network training & model selection

Training is done in Python 2.7 using Keras deep learning library [71] and tools from Scikit-Learn [72] on a computer node equipped with Tesla K40 GPU, two Intel Xeon E5-2620-v2 CPUs with a base frequency of 2.1 GHz and 32 GB main memory, thanks to CSC-IT Center for Science, Finland.

As features, 30 MFCCs are extracted each from the DSB output and the signal from background facing microphone; 19 LPCs are also extracted from the DSB output after pre-emphasis. The rationale behind the choice of these features is

in [73] where a thorough analysis of incremental improvements brought in by the combinations of features for TR-real data of CHiME-3 challenge is studied for a FFNN-based TF mask prediction.

The mini-batch size for Adam optimizer is 128 sequences with a learning rate of 10^{-4}. The default values of $\beta_1 = 0.9$ and $\beta_2 = 0.999$ are used, which are the parameters needed by the optimizer [29]. The mask approximation objective function is employed — MSE between the ground-truth and predicted masks is monitored while training. An early stopping patience of 30 epochs is chosen empirically; the model weights corresponding to that epoch where the improvement had just begun to stop are chosen as the final weights.

A naive grid-search is performed on the number of hidden layers and number of LSTM cells per hidden layer — a maximum of 3 hidden layers with number of LSTM cells varied among $32, 64, 128, 256$ and 512 respectively, every other hidden layer being of the same sizes. Preliminary experiments on the sequence length suggested a sequence length of $T = 100$ time frames, longer sequences not helping any further or yielding sub-par validation MSEs. Both the $tanh$ and $softsign$ — as activations for LSTM units — resulted in very similar performance.

4.4 Baseline models

The FFNN based post-filter framework, for comparison, uses seven adjacent frames for each frame to include context in the prediction. The model consists of two hidden layers, with 600 units per layer, and ℓ_2-regularization of $1.6 \cdot 10^{-4}$. As in the previous case, grid search — with 7.5% of TR data used for cross-validation — is applied for hyperparameter selection. The activations for hidden layer and output layer are $tanh()$ and linear respectively. The best performing model for the cross-validation data is selected from 50 first epochs. A similar ensemble of 5 networks is also constructed for fair comparison.

Both the RNN and FFNN based beamforming + post-filtering approaches are evaluated against the baseline MVDR beamforming provided by the CHiME-3 Challenge organizers. The beamformer obtains speaker position cues from an SRP-PHAT pseudo-spectrum and uses a multi-channel covariance matrix estimated from 400 ms to 800 ms of context prior to the start of utterances for the computation of its weights [1].

As mentioned previously, the work for this thesis commenced from TUT's participation in the CHiME-3 challenge. The submission (see Appendix) employed a similar DSB followed by post-filter framework but with more sophisticated methods

connected in a slightly ad-hoc fashion. A speech-weighted variation was suggested to the traditional steered response power (SRP) phase transform (SRP-PHAT) for estimating target speaker's direction. A Kalman filter based tracking was included to obtain smooth trajectory of speaker's state and avoid possible outliers. The framework also employed an MLP to classify microphone signal frames into 3 classes: speech, speech in background and background respectively. The MLP post-filter used spatial features extracted for both the target and background, MFCCs from DSB output and backward facing microphone and did not include LPCs. The Wiener filter was the target TF mask whose values were μ-law transformed while training — the predicted TF masks were later inverse μ-law transformed.

4.5 ASR backend

The baseline ASR backend provided by the CHiME-3 Challenge organizers is used as such [1] — the supplied back-end was state-of-the-art at the time when the challenge was announced. It consists of a 7-layer DNN with 2048 hidden neurons in each layer. The input layer uses 5 frames of left and right context, totaling to 440 units. The DNN is pre-trained with a restricted Boltzmann machine (RBM), followed by cross-entropy training and finally a sequence discriminative training.

Running the above baseline requires greater computational resources and hence scarcely used. Instead, a traditional GMM-HMM recognizer was used to obtain a preliminary glimpse of what can be expected out of the sophisticated DNN-based ASR. The details of the GMM-HMM recognizer can be found in [1].

4.6 Results and discussion

The best performing model, as monitored by the validation MSE, consists of 2 hidden BLSTM layers with 256 LSTM cells each for forward and backward directions. Other complex models did not improve the performance any further. Figure 4.1 gives a visual glimpse of mask prediction capability of the BLSTM network. Throughout, unless specifically mentioned, the results are shown for an ensemble of networks as the ensemble always resulted in lesser MSE scores.

The objective quality and intelligibility scores for the separated signals, obtained by BLSTM networks predicting Wiener and log-ratio masks are given in figure 4.2. It can be observed that predicting log-ratio masks unanimously resulted in better scores across all datasets. While there is only very minuscule difference in the STOI scores in simulated settings, there is a consistent difference of 0.02 units in real settings. The SNR_{fw} scores show a big difference of approximately 1 dB, especially

in real environments. The better performance obtained by predicting log-ratio mask can be attributed to the fact that background noise estimation is not required as the mask formulation uses only clean signal and the mixture; inaccuracies in the noise estimation have resulted in reduction in efficient noise suppression in the case of Wiener filters.

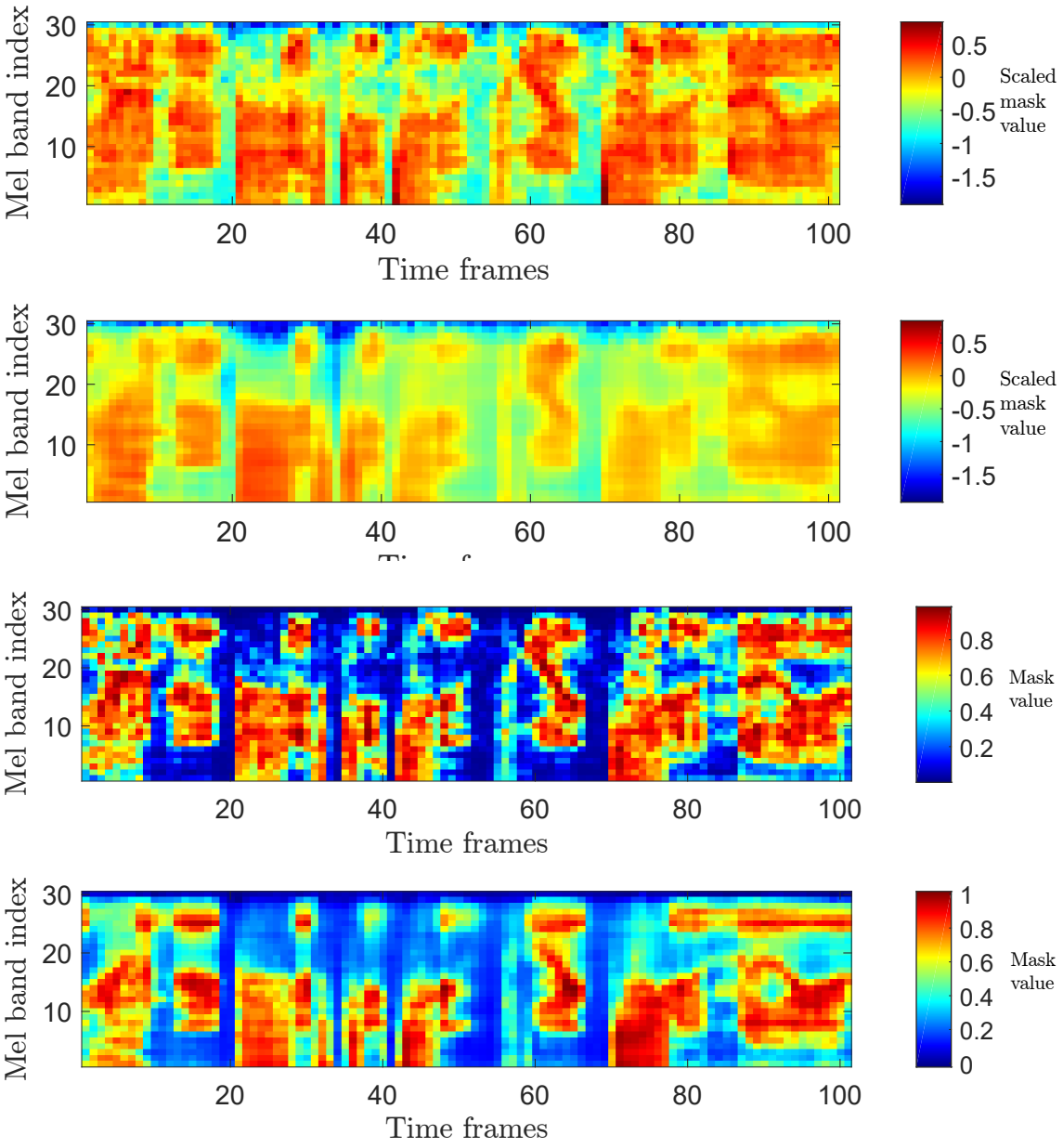


Figure 4.1 *Prediction by a BLSTM ensemble illustrated for part of an utterance from DT set. The predicted masks from the individual networks of the ensemble are optimal in a mean squared error (MSE) sense.* ***First row****: Ground-truth log-ratio mask;* ***Second row****: Predicted log-ratio mask;* ***Third row****: Ground-truth Wiener mask;* ***Fourth row****: Predicted Wiener mask*

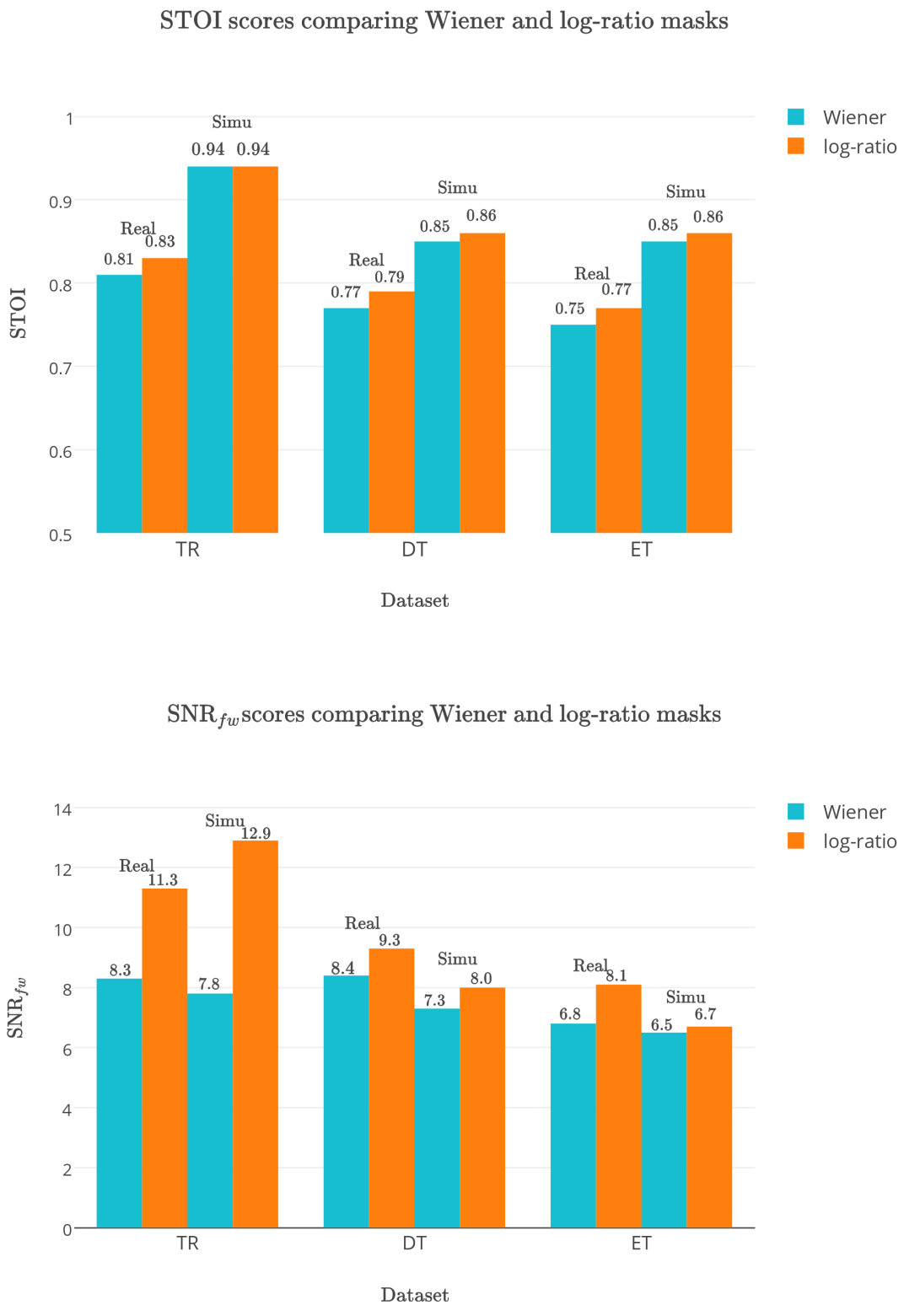


Figure 4.2 *Objective scores for speech separation accomplished by predicting Wiener filter and log-ratio post-filter TF masks for CHiME-3 data*

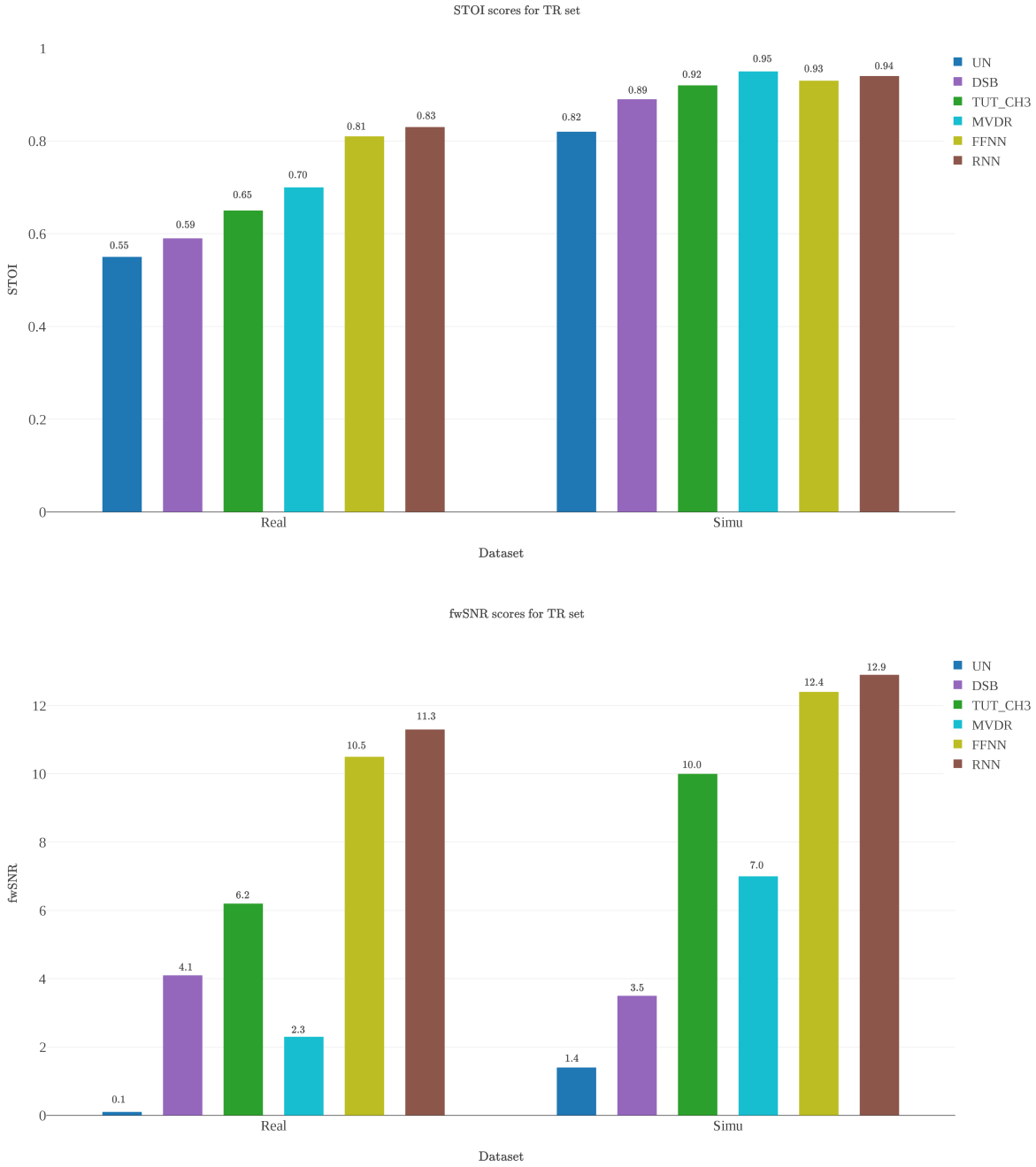


Figure 4.3 *Objective scores for speech separation accomplished by predicting log-ratio post-filter TF masks: comparison of proposed (RNN-BLSTM) method against other methods for CHiME-3 TR data. Description of the legend: UN refers to using unprocessed (noisy) speech directly for evaluation; DSB refers to delay-and-sum beamforming without any post-filtering; MVDR is the baseline method provided by CHiME-3 organizers; TUT_CH3 is based on TUT's contribution to the CHiME-3 challenge; FFNN refers to DSB followed by FFNN based post-filter; RNN refers to DSB followed by RNN-BLSTM based post-filter*

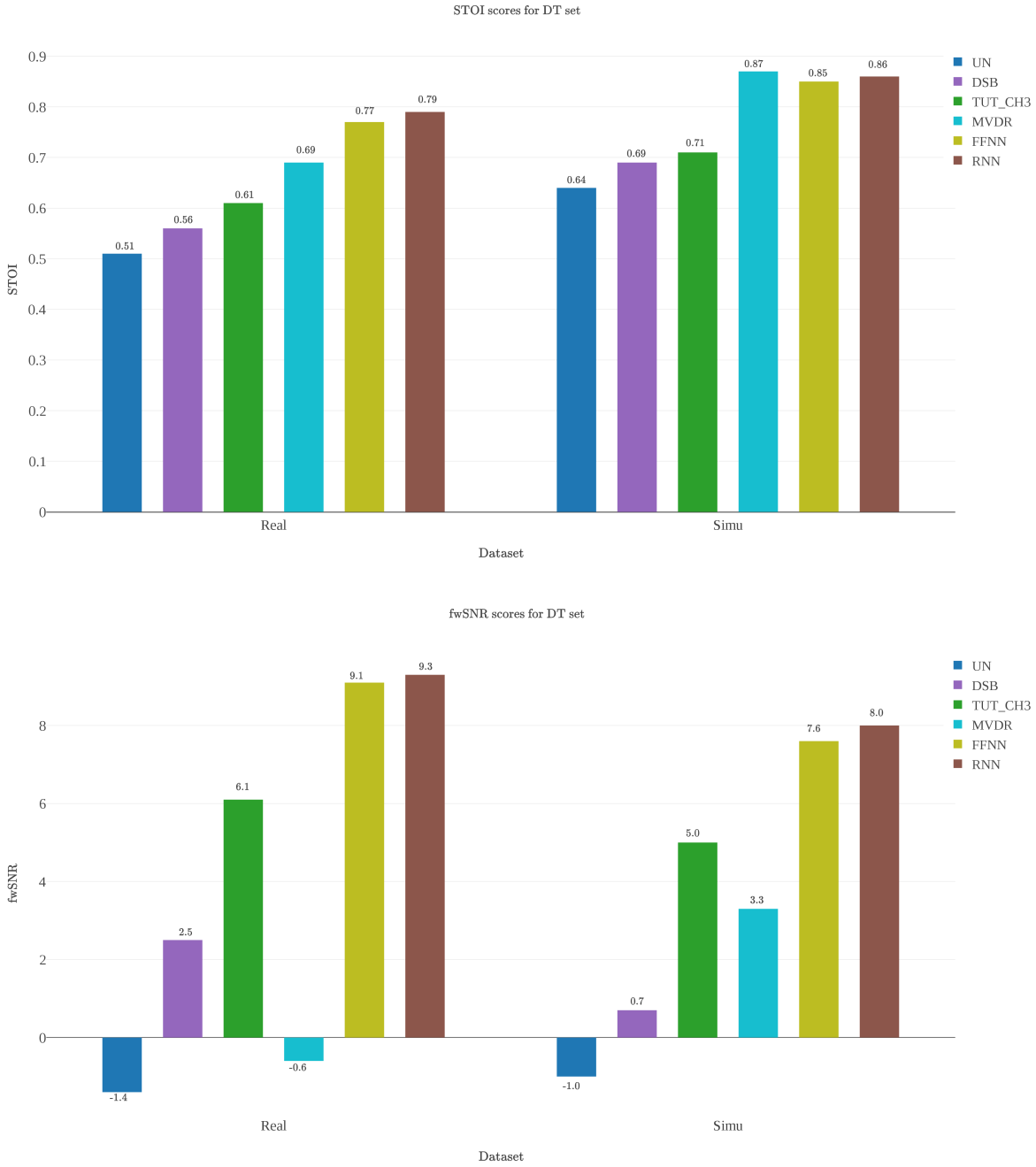


Figure 4.4 *Objective scores for speech separation accomplished by predicting log-ratio post-filter TF masks: comparison of proposed (RNN) method against other methods for CHiME-3 DT data.*

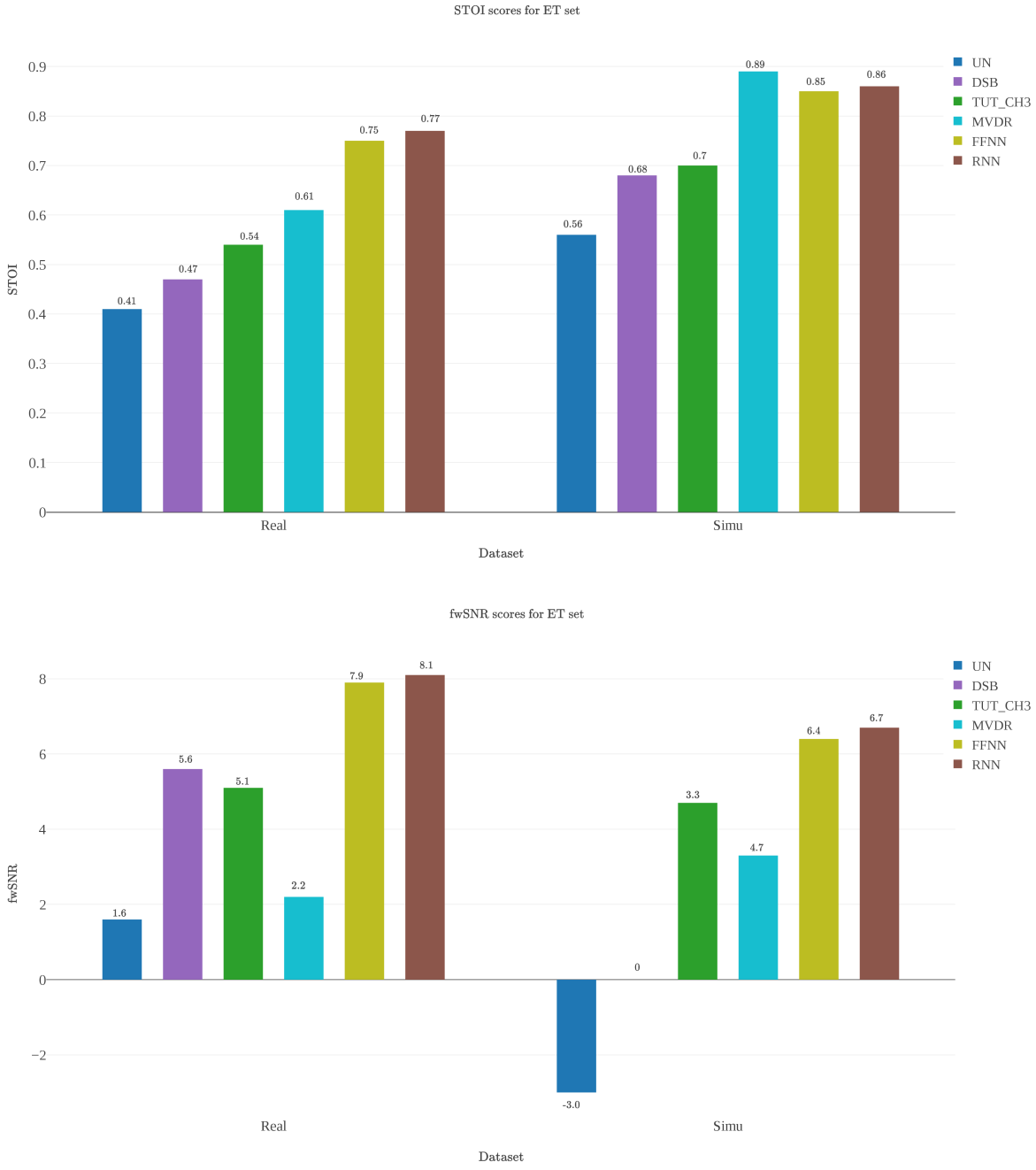


Figure 4.5 *Objective scores for speech separation accomplished by predicting log-ratio post-filter TF masks: comparison of proposed (RNN) method against other methods for CHiME-3 ET data.*

The objective scores for the BLSTM based post-filter predicting log-ratio masks are compared against other baselines in figures 4.3, 4.4 and 4.5. Across all datasets, in real settings, BLSTM performs the best in terms of STOI measure, with a con-

sistent 0.02 units improvement compared to FFNN in TR, DT and ET datasets; in simulated settings on the other hand, MVDR seems to outperform. Concerning the fwSNR scores, BLSTM again takes the lead in both real and simulated settings — in real settings, there is at least 0.2 dB improvement compared to that FFNN, while the performance of MVDR is inferior. The use of BLSTM layers in the network has aided in providing more time context resulting in better mask prediction. It is worthwhile to note that the fwSNR score of MVDR output is especially abysmal in real settings, compared even to the simpler DSB presumably due to inaccuracies in the estimates of noise covariance matrix in real settings. Thus the MVDR's overly optimistic performance in simulated settings is once again re-iterated.

Throughout all the results, there is a consistent decline in the performance in DT and ET sets due to one or more mismatch problems — speaker, environment or noise — which can be addressed by adaptation techniques in the context of speech separation.

The beamformed signals, on post-filtering with a single BLSTM and fed into the DNN-based ASR, resulted in a WER performance depicted in Table 4.2. Table 4.3 reveals the performance of the ensemble, surprisingly showing sub-par results compared to a single BLSTM network despite showing better objective scores. In both the cases, there is a big gap between the performances in DT and ET sets, with around 10% difference in WERs. There is a significant improvement in observed WERs compared to TUT's submission to CHiME-3 challenge (see Appendix) — with WER reduced by around 1.8% in DT (real), 2% in DT (simu), 3% in both ET (real) and ET (simu) datasets respectively (comparisons made against the non-ensemble version of BLSTM-based mask prediction).

On interpreting the objective scores against WERs for real and simulated data in general (see Figure 4.6), despite significantly better STOI scores — in both DT and ET datasets —, WER in simulated scenario is worse than in real scenario; but fwSNR scores in simulated scenario are consistently lower in comparison.

The discussion in this paragraph and the following one is regarding the real scenario; please refer to Figure 4.7. Of the four environments — both in DT and ET sets — the bus environment seems to be the most challenging with the highest WER; this can perhaps be attributed to the more diffuse nature of noise as well as unreliable localization of the target due to speaker movement. An interesting observation in the DT set is that, though the objective scores — both STOI and fwSNR — for the separated speech in street environment are the worst, the WER is roughly better by 1.86% compared to that of bus environment. The WERs in other

environments — bus, cafeteria and pedestrian — show a consistent relationship with the objective scores; with improvements in STOI and fwSNR scores, the WERs improves.

Table 4.2 *WER scores from a DNN based ASR for speech separated by a single BLSTM network predicting log-ratio mask*

	BUS	CAF	PED	STR	**AVG**
DT(Real)	10.6%	8.04%	7.14%	9.39%	**8.79%**
DT(Simu)	9.42%	13.85%	9.90%	12.99%	**11.54%**
ET(Real)	22.66%	16.01%	19.41%	13.97%	**18.01%**
ET(Simu)	15.13%	19.84%	22.15%	27.06%	**21.05%**

Table 4.3 *WER scores from a DNN based ASR for speech separated by an ensemble of BLSTM networks predicting log-ratio mask. Despite yielding smaller prediction MSEs and better objective scores compared to a single BLSTM network, the ensemble results in sub-par WER performance.*

	BUS	CAF	PED	STR	**AVG**
DT(Real)	11.0%	8.63%	6.92%	9.14%	**8.92%**
DT(Simu)	9.79%	14.75%	9.93%	14.68%	**12.29%**
ET(Real)	24.51%	17.73%	19.23%	14.14%	**18.90%**
ET(Simu)	16.27%	21.76%	23.76%	31.02%	**23.20%**

Both the objective scores and WERs reveal the clear mismatch between the DT and ET datasets, the performance of separation being the worst across all the environments of ET dataset. The pedestrian environment which exhibited the least WER and the best STOI scores in DT set stands second worst in the ET set. The street environment exhibits a similar anomaly in the ET set as well.

The simulated scenario exhibits complete reversal of trends compared to real scenario — refer to Figure 4.8. The WER — both in DT and ET datasets — is the least in bus environment. The DT dataset reveals lots of inconsistencies: both the STOI and fwSNR scores are slightly better in pedestrian environment compared to bus environment, yet the WER is slightly higher in comparison; despite the same STOI scores and a slightly better fwSNR score in the cafeteria environment compared to the street, its WER is slightly higher. Within the ET set the WERs are consistent with the objective scores. It is noteworthy to observe that fwSNR seems to have an upper hand in predicting the WER performance of ET-simu dataset; for

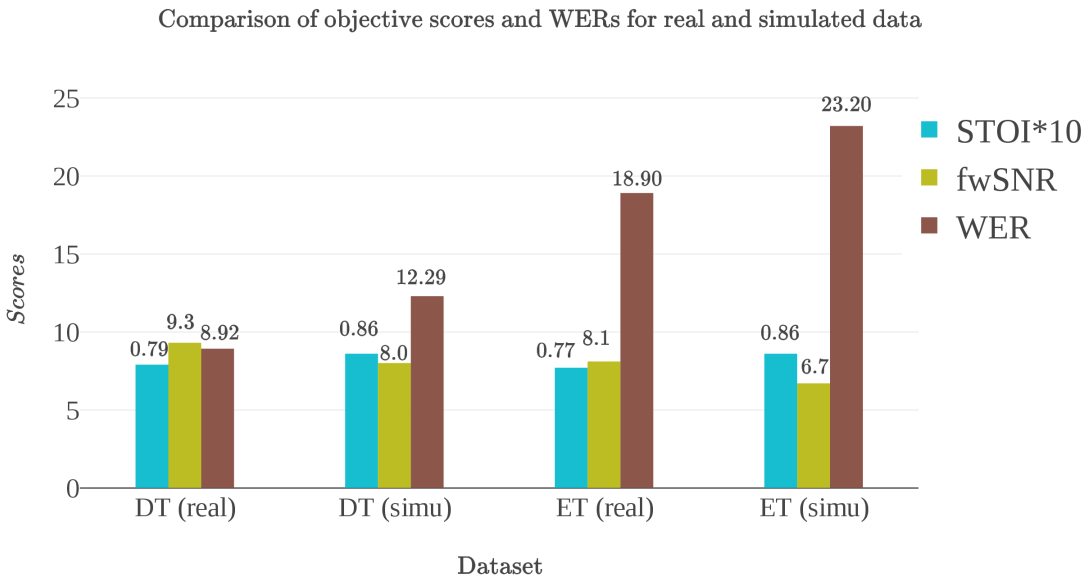


Figure 4.6 *Objective scores and WERs for CHiME-3 real and simulated data*

example, despite better STOI scores compared to ET-real dataset, with an exception of bus environment, the WERs in other environments are very high in comparison. Nevertheless, despite all these analysis, the ability of the objective scores in predicting the actual machine intelligibility is still unclear and demands more refined analysis.

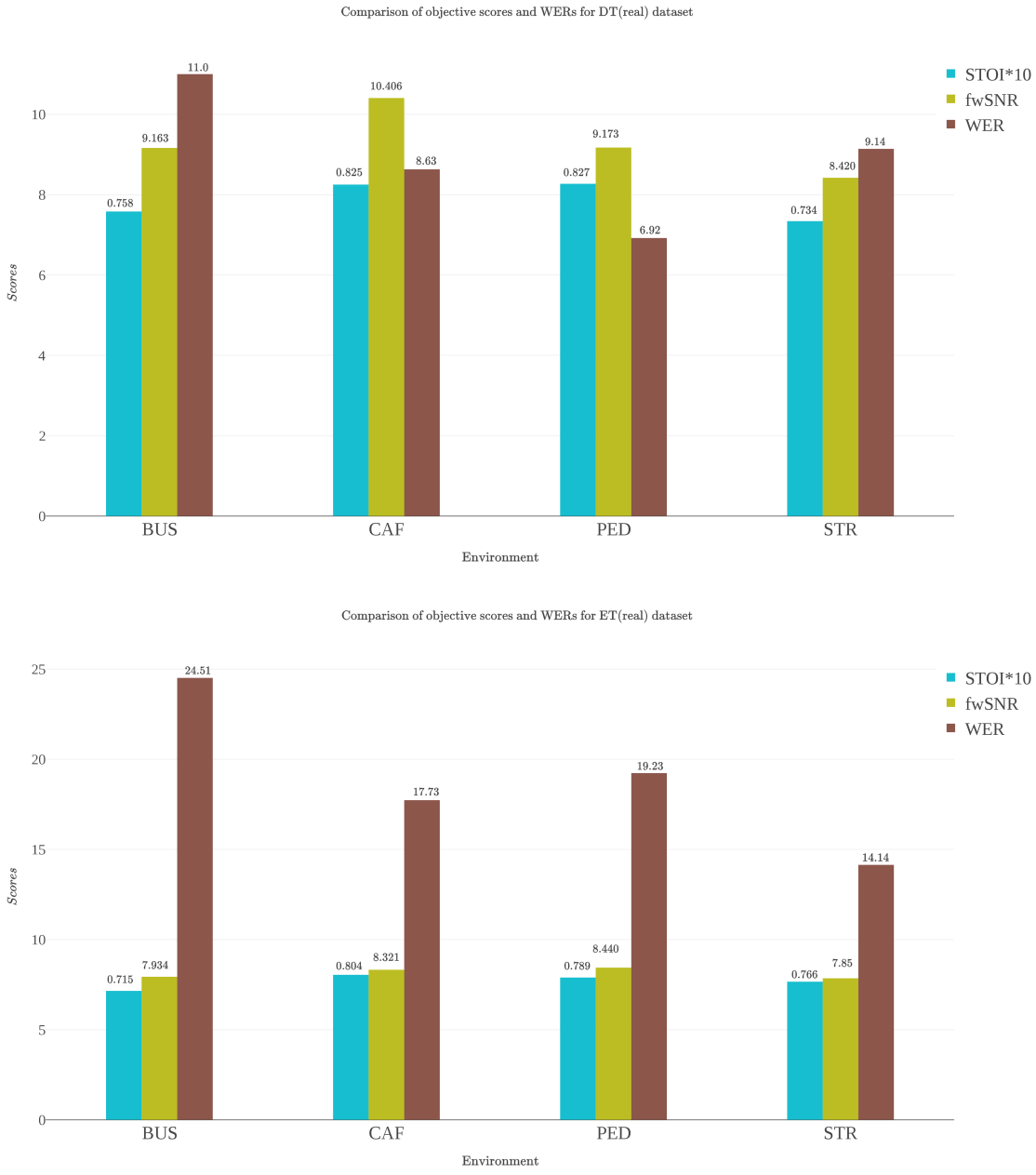


Figure 4.7 *Environment-wise objective scores and WERs for CHiME-3 real data*

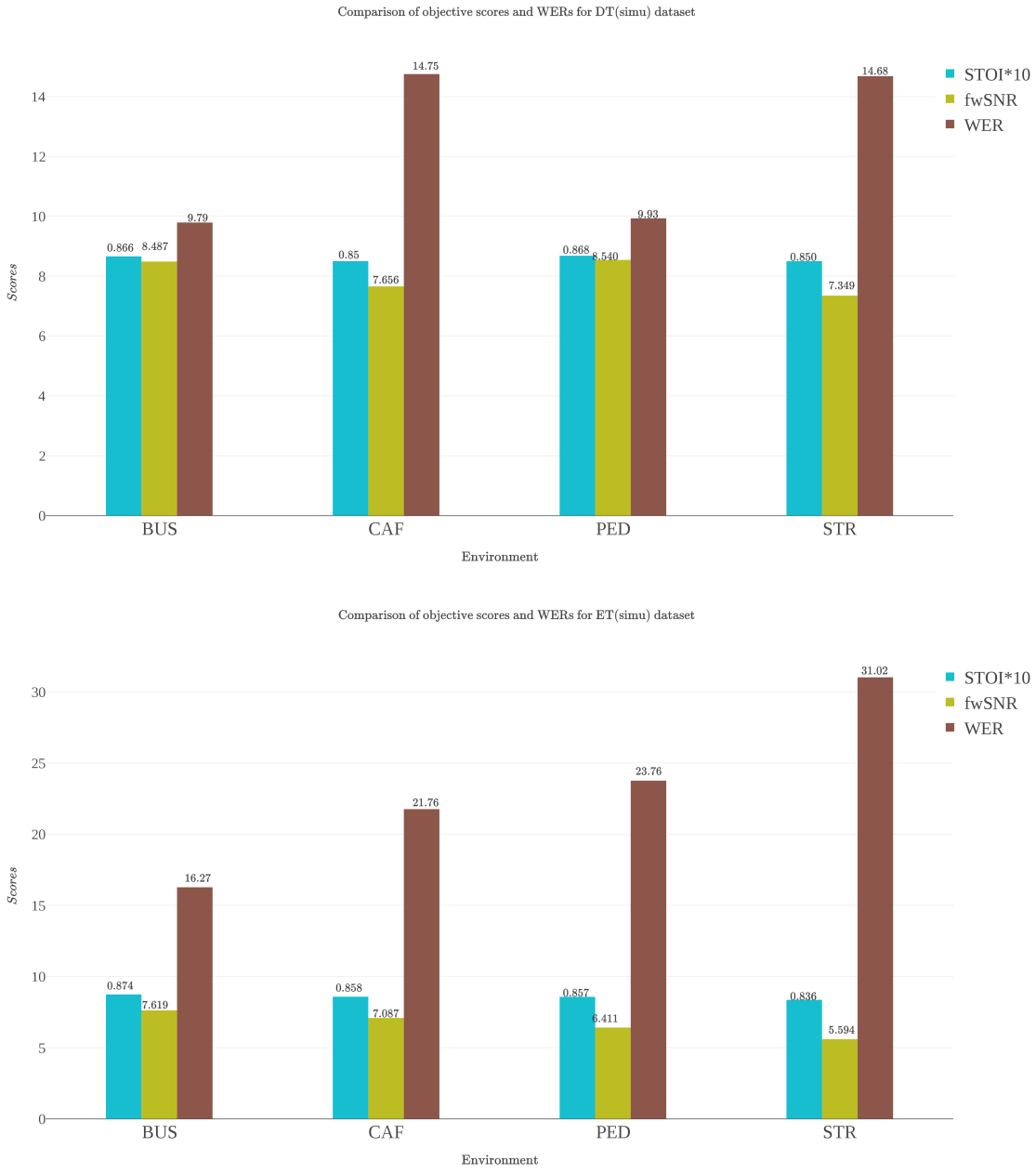


Figure 4.8 *Environment-wise objective scores and WERs for CHiME-3 simulated data*

5. CONCLUSION AND FUTURE DIRECTIONS

In this work we proposed the application of recurrent neural networks for supervised speech separation using multiple microphones in challenging real-world environments. A BLSTM RNN model predicting TF masks was tailored to be used as a post-filter for a delay-and-sum beamformer. TF mask prediction is naturally a sequential data prediction problem and the use of RNNs can be justified.

Spectral features representing the speech signal and background noise along with phase based features extracted from TDoA values were used to train the RNN to predict the T-F mask. Mask prediction training was done to minimize the MSE between the true and predicted masks. The method was applied to CHiME-3 challenge data and two variants of T-F masks — Wiener filter and log-ratio amplitude mask —were investigated.

The quality and intelligibility of the separated speech signals were reported using objective measures. The proposed separation framework was also applied as a naive front-end to a DNN-based ASR system and the resulting WERs were reported. The results supported the use of RNNs when compared against a FFNN framework and also a baseline MVDR beamformer.

While the results are promising, there are several scopes for improvement and a plethora of open questions to be addressed. As immediate extensions to the developed framework, signal approximation approach to TF mask prediction can be tried by optimizing a speech separation objective directly with mask prediction as an intermediate objective. The DSB beamformer can be replaced by another sophisticated beamformer.

A more refined way of performing hyperparameter search has to be explored. Convolutional neural networks (CNNs) are in vogue and have tasted immense success in speech community as well [74]. LSTMs can be replaced or used in conjunction with CNNs for TF mask prediction. While the current ensemble framework used invariably the same data but with randomly initialized networks, a better ensemble can be developed, something akin to stacking as explored in [75]. Adaptation techniques can be explored in the context of speech separation to address the mismatch problems.

A thorough analysis on whether — and how — speech intelligibility/quality assessment translates to improvements in WERs should be done. Instead of applying speech separation as a naive front end to ASR, alignment information from ASR— for example obtained from a rough pass of noisy mixture through the ASR — can be incorporated as a feature in the separation framework. This can also be further extended by a joint training of separation front-end and ASR back-end, as they have a symbiotic relationship [76].

BIBLIOGRAPHY

[1] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third CHiME speech separation and recognition challenge: Dataset, task and baselines," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 504–511, 2015.

[2] J. Treichler, "Signal processing: A view of the future, Part 1 [Exploratory DSP]," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 118–118, 2009.

[3] R. Lyon, "Machine Hearing: An Emerging Field [Exploratory DSP," *IEEE Signal Processing Magazine*, vol. 27, no. 5, pp. 131–139, 2010.

[4] A. Hurmalainen, "Robust Speech Recognition with Spectrogram Factorisation," *PhD thesis, Tampere University of Technology*, 2014.

[5] W. Xiong, J. Droppo, X. Huang, F. Seide, M. L. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Toward Human Parity in Conversational Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2410–2423, 2017.

[6] J. Smith, "Spectral Audio Signal Processing." `http://ccrma.stanford.edu/~jos/sasp/`, 2011. Accessed: 10-04-2017.

[7] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 708–712, IEEE, 2015.

[8] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *Computing Research Repository, (CoRR)*, vol. abs/1708.07524, 2017.

[9] P. Pertilä, "Online blind speech separation using multiple acoustic speaker tracking and time-frequency masking," *Computer Speech and Language*, vol. 27, no. 3, pp. 683–702, 2013.

[10] K. Kondo, "Speech Quality," in *Subjective Quality Measurement of Speech*, pp. 7–20, Springer, 2012.

[11] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[12] S. Rickard and O. Yilmaz, "On the approximate W-disjoint orthogonality of speech," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. I–529–I–532, 2002.

[13] Yuxuan Wang, A. Narayanan, and DeLiang Wang, "On Training Targets for Supervised Speech Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

[14] D. Wang, "On Ideal Binary Mask As the Computational Goal of Auditory Scene Analysis," in *Speech Separation by Humans and Machines*, pp. 181–197, Kluwer Academic Publishers, 2005.

[15] A. S. Bregman, *Auditory scene analysis: the perceptual organization of sound*. MIT Press, 1990.

[16] S. Haykin and Z. Chen, "The Cocktail Party Problem," *Neural Computation*, vol. 17, no. 9, pp. 1875–1902, 2005.

[17] N. Li and P. C. Loizou, "Factors influencing intelligibility of ideal binary-masked speech: Implications for noise reduction," *The Journal of the Acoustical Society of America*, vol. 123, no. 3, p. 1673, 2008.

[18] D. Wang, U. Kjems, M. S. Pedersen, J. B. Boldt, and T. Lunner, "Speech intelligibility in background noise with ideal binary time-frequency masking," *The Journal of the Acoustical Society of America*, vol. 125, no. 4, pp. 2336–2347, 2009.

[19] S. Srinivasan, N. Roman, and D. Wang, "Binary and ratio time-frequency masks for robust speech recognition," *Speech Communication*, vol. 48, no. 11, pp. 1486–1501, 2006.

[20] C. Hummersone, T. Stokes, and T. Brookes, "On the Ideal Ratio Mask as the Goal of Computational Auditory Scene Analysis," in *Blind Source Separation: Advances in Theory, Algorithms and Applications*, pp. 349–368, Springer Berlin Heidelberg, 2014.

[21] J. Benesty, J. Chen, and Y. Huang, "Conventional Beamforming Techniques," in *Microphone Array Signal Processing*, pp. 39–65, Springer, 2008.

[22] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.

[23] N. Ito, N. Ono, E. Vincent, and S. Sagayama, "Designing the Wiener post-filter for diffuse noise suppression using imaginary parts of inter-channel cross-spectra," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2818–2821, 2010.

[24] K. Kumatani, J. McDonough, and B. Raj, "Microphone Array Processing for Distant Speech Recognition: From Close-Talking Microphones to Far-Field Sensors," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 127–140, 2012.

[25] R. Zelinski, "A microphone array with adaptive post-filtering for noise reduction in reverberant rooms," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2578–2581, 1988.

[26] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning From Data*. AMLBook, 2012.

[27] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[28] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *Computational Research Repository (CoRR)*, vol. abs/1212.5701, 2012.

[29] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *Computation Research Repository, (CoRR)*, vol. abs/1412.6980, 2014.

[30] Y. Bengio, P. Frasconi, and P. Simard, "The problem of learning long-term dependencies in recurrent networks," in *IEEE International Conference on Neural Networks*, pp. 1183–1188, 1993.

[31] R. Pascanu, T. Mikolov, and Y. Bengio, "On the Difficulty of Training Recurrent Neural Networks," in *Proceedings of the 30th International Conference on International Conference on Machine Learning (ICML)*, pp. III–1310–III–1318, Journal of Machine Learning Research (JMLR), 2013.

[32] F. Gers, "Long Short-Term Memory in Recurrent Neural Networks," *PhD thesis, Ecolè Polytechniquè Fèdèrale De Lausanne*, 2001.

[33] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *Computational Research Repository (CoRR)*, vol. abs/1406.1078, 2014.

[34] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A Critical Review of Recurrent Neural Networks for Sequence Learning," *Computing Research Repository (CoRR)*, vol. abs/1506.00019, 2015.

[36] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[37] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, J. Chen, J. Chen, Z. Chen, M. Chrzanowski, A. Coates, G. Diamos, K. Ding, N. Du, E. Elsen, J. Engel, W. Fang, L. Fan, C. Fougner, L. Gao, C. Gong, A. Hannun, T. Han, L. V. Johannes, B. Jiang, C. Ju, B. Jun, P. LeGresley, L. Lin, J. Liu, Y. Liu, W. Li, X. Li, D. Ma, S. Narang, A. Ng, S. Ozair, Y. Peng, R. Prenger, S. Qian, Z. Quan, J. Raiman, V. Rao, S. Satheesh, D. Seetapun, S. Sengupta, K. Srinet, A. Sriram, H. Tang, L. Tang, C. Wang, J. Wang, K. Wang, Y. Wang, Z. Wang, Z. Wang, S. Wu, L. Wei, B. Xiao, W. Xie, Y. Xie, D. Yogatama, B. Yuan, J. Zhan, and Z. Zhu, "Deep Speech 2: End-to-end Speech Recognition in English and Mandarin," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 173–182, JMLR.org, 2016.

[38] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware Neural Language Models," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 2741–2749, 2016.

[39] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 86–96, Association for Computational Linguistics (ACL), 2016.

[40] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 652–663, 2017.

[41] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation and active learning," in *Proceedings of the 7th International Conference on Neural Information Processing Systems*, pp. 231–238, MIT Press, 1994.

[42] M. P. Perrone and L. N. Cooper, "When Networks Disagree: Ensemble Methods for Hybrid Neural Networks," in *Neural networks for speech and image processing*, pp. 126–142, Chapman and Hall, 1993.

[43] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "A Regression Approach to Speech Enhancement Based on Deep Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2015.

[44] A. Narayanan and D. Wang, "Investigation of Speech Separation as a Front-End for Noise Robust Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 826–835, 2014.

[45] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1562–1566, 2014.

[46] K. Han and D. Wang, "A classification based approach to speech segregation," *The Journal of the Acoustical Society of America*, vol. 132, no. 5, p. 3475, 2012.

[47] G. Kim, Y. Lu, Y. Hu, and P. C. Loizou, "An algorithm that improves speech intelligibility in noise for normal-hearing listeners.," *The Journal of the Acoustical Society of America*, vol. 126, pp. 1486–94, 9 2009.

[48] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Speech Enhancement with LSTM Recurrent Neural Networks and its Application to Noise-Robust ASR," in *Proceedings of the 12th International Conference on Latent Variable Analysis and Signal Separation - Volume 9237*, pp. 91–99, Springer-Verlag New York, Inc., 2015.

[49] P. Pertilä and J. Nikunen, "Distant speech separation using predicted time-frequency masks from spatial features," *Speech Communication*, vol. 68, pp. 97–106, 2015.

[50] J. R. Hershey, S. J. Rennie, P. A. Olsen, and T. T. Kristjansson, "Super-human multi-talker speech recognition: A graphical modeling approach," *Computer Speech & Language*, vol. 24, no. 1, pp. 45–66, 2010.

[51] F. Weninger, J. R. Hershey, J. Le Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 577–581, 2014.

[52] Z. Jin and D. Wang, "A Supervised Learning Approach to Monaural Segregation of Reverberant Speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007.

[53] G. Kim, Y. Lu, Y. Hu, and P. C. Loizou, "An algorithm that improves speech intelligibility in noise for normal-hearing listeners.," *The Journal of the Acoustical Society of America*, vol. 126, no. 3, pp. 1486–94, 2009.

[54] Yuxuan Wang and DeLiang Wang, "Towards Scaling Up Classification-Based Speech Separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1381–1390, 2013.

[55] D. S. Williamson, Y. Wang, and D. Wang, "Complex Ratio Masking for Monaural Speech Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 3, pp. 483–492, 2016.

[56] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, 1994.

[57] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An Overview of Noise-Robust Automatic Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[58] J. Droppo, "Feature Compensation," in *Techniques for Noise Robustness in Automatic Speech Recognition*, pp. 229–250, John Wiley and Sons Ltd, 2012.

[59] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech & Language*, vol. 12, no. 2, pp. 75–98, 1998.

[60] R. Józefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the Limits of Language Modeling," *Computing Research Repository, (CoRR)*, vol. abs/1602.02410, 2016.

[61] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, and M. Bacchiani, "Factored Spatial and Spectral Multichannel Raw Waveform CLDNNs," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016.

[62] Y. Miao, H. Zhang, and F. Metze, "Speaker Adaptive Training of Deep Neural Network Acoustic Models Using i-Vectors," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 11, pp. 1938–1949, 2015.

[63] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third CHiME speech separation and recognition challenge: Analysis and outcomes," *Computer Speech and Language*, vol. 46, 2017.

[64] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. Green, "The PASCAL CHiME speech separation and recognition challenge," *Computer Speech and Language*, vol. 27, no. 3, pp. 621–633, 2013.

[65] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'CHiME' Speech Separation and Recognition Challenge: An overview of challenge systems and outcomes," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2013.

[66] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third CHiME speech separation and recognition challenge: Dataset, task and baselines," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 504–511, 2015.

[67] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An Analysis of Environment, Microphone and Data Simulation Mismatches in Robust Speech Recognition," *Computer Speech and Language*, vol. 46, no. C, pp. 535–557, 2017.

[68] P. Pertilä and J. Nikunen, "Microphone Array Post-Filtering Using Supervised Machine Learning for Speech Enhancement," in *INTERSPEECH*, 2014.

[69] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4214–4217, 2010.

[70] Y. Hu and P. C. Loizou, "Evaluation of Objective Quality Measures for Speech Enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 229–238, 2008.

[71] François Chollet, "Keras Documentation." `https://keras.io/`. Accessed: 10-04-2017.

[72] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[73] P. Pertilä, "Microphone-Array-Based Speech Enhancement Using Neural Networks," in *Parametric Time-Frequency Domain Spatial Audio*, pp. 291–325, Wiley, 2017.

[74] L. Hui, M. Cai, C. Guo, L. He, W. Q. Zhang, and J. Liu, "Convolutional maxout neural networks for speech separation," in *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 24–27, 2015.

[75] J. L. Roux, S. Watanabe, and J. R. Hershey, "Ensemble learning for speech enhancement," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 1–4, 2013.

[76] Z. Q. Wang and D. Wang, "A Joint Training Framework for Robust Automatic Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 796–806, 2016.

APPENDIX: TECHNICAL REPORT OF TUT'S SUBMISSION TO CHIME-3 CHALLENGE

AUTOMATIC SPEECH RECOGNITION WITH MULTICHANNEL NEURAL NETWORK BASED SPEECH ENHANCEMENT

Pasi Pertilä, Antti Hurmalainen, Shriram Nandakumar, Tuomas Virtanen

Tampere University of Technology, Department of Signal Processing

ABSTRACT

We investigate the enhancement of speech signals captured by a mobile multi-microphone device in different noisy real surroundings. An estimate of the speaker direction among competing sources is obtained by a steered response power weighted by speech likelihood. This allows the use of single source tracking to provide a smoothed speaker trajectory, which is used to steer a delay and sum beamformer. A novel neural network based approach is used to predict a Wiener filter to further enhance the beamformed speech signal. The neural network uses both amplitude and phase based features and temporal context. Objective signal quality metrics show that the post-filtering scheme improves the speech quality over beamforming. The system is used as a front-end of automatic speech recognition in the 3rd CHiME Challenge, reducing the word error rate of real-world multichannel recordings by more than a third using state-of-the-art back-ends.

Index Terms— Speech enhancement, Automatic speech recognition, Microphone arrays, Multilayer perceptrons, Kalman filters

1. INTRODUCTION

Robust processing of speech remains a topic of persisting or even increasing interest as speech-operated interfaces are gradually brought into a large variety of mobile devices, including wearable technology with no conventional touch-operated interface in the first place. For reliable operation and thus broad adoption, the devices should be usable in a wide range of everyday environments featuring diverse types of noise and signal degradation [1].

The increased attention to robustness has resulted in major advances in the scope and effectiveness of automatic speech recognition (ASR) algorithms [2, 3]. A few noteworthy approaches to enhancement and recognition of conventional single-channel speech include auditory-inspired features [4], source separation algorithms such as non-negative matrix factorisation (NMF) [5, 6], and deep neural networks (DNNs) [7, 8, 9, 10]. Hybrid solutions, combining several methods at multiple levels from front-end enhancement to back-end decoding, have produced promising results in standardized evaluations [11, 12]. Meanwhile, as multi-microphone devices are becoming the norm, spatial information from multichannel recordings is often available, providing another potentially significant component for source separation and target signal enhancement.

Time-frequency (TF) domain separation methods multiply the observed noisy spectrogram with a mask to separate the desired signal. The mask is either binary or real valued, and obtaining the ideal binary mask (IBM) is widely considered as the goal of computational auditory scene analysis (CASA) [13]. More recently, it has been argued that the real valued ideal ratio mask (IRM) [14] may be more closely related to auditory processes than IBM suggested by certain ASR and speech intelligibility measurements [15]. Obtaining the IBM is a binary classification problem, whereas obtaining the IRM can be seen as regression. In single-channel cases, a few proposed mask learning algorithms include NMF [16], amplitude modulation spectrograms (AMSs) [17], and support vector machines (SVMs) [18], whereas stereo based enhancement methods can take advantage of spatial cues, i.e., interaural time and level differences (ITD, ILD) [14, 19, 20, 21, 22, 23].

Beamforming refers to a linear combination of multichannel data to enhance and/or cancel spatial directions. The approach requires knowledge of microphone positions and source directions that must must be known or otherwise estimated. The steered response power (SRP) method can be used to evaluate the likelihoods of different angles of sound arrival [24], and the phase alignment transform (SRP-PHAT) variant has been found robust in room environments [25]. Tracking algorithms such as Kalman filters and particle filters provide a smoothed trajectory of a source by filtering the sequential observations. In order to track speech sources, voice activity detection (VAD) scheme can be integrated to handle natural pauses in speech [26]. In case of multiple speakers, pitch cues distinguish direction estimates between speakers [27]. In general, tracking of multiple sources is more complicated than single source tracking, with issues such as data-association to be resolved [28].

ASR for multichannel audio using DNNs and beamforming was investigated in [29], where concatenating single channel amplitude features was preferred over the output of a delay and sum beamformer (DSB). Recently, multichannel spatial cues were proposed for the IRM predicting with a multilayer perceptron (MLP) for regression [30]. The method

predicts the Wiener filter (WF), a special case of IRM, from phase-difference based features between microphone array channels and used simulations to create training samples. The predicted WF which has values in the range of $[0, 1]$ is then applied to the output of a beamformer as a post-filter. The method was extended for separation of multiple speech sources in [31] exploiting the TF sparseness of speech.

This paper extends the mask prediction of [30] to realistic use cases of mobile devices, and evaluates its performance for ASR. A VAD type approach is used to produce a speech weighted steered response power. This allows to obtain the direction of a semi-static speaker over otherwise dominating non-speech and noisy speech directions. Therefore, the potential multiple source tracking scenario is reduced into tracking of a single source, which is a simpler problem and thus more robust to solve. Several other improvements are introduced over [30], such as using a large set of over 8000 sentences for the post-filter training, using both amplitude spectrum and temporal information in addition to the phase only based features in post-filter prediction, and performing nonlinear transformation of Wiener filter values before learning to predict them with regression in order to emphasize the noise attenuation capability of the post-filter.

The method is applied to the 3rd CHiME Challenge data [32] comprising real and simulated multi-microphone recordings of Wall Street Journal speech over everyday noise environments, and evaluated in terms of objective speech quality metrics, as well as noise-robust ASR accuracy.

The paper structure is the following. Section 2 describes the multichannel speech enhancement method that is used to provide the enhanced signals for ASR. Section 3 describes the CHiME3 challenge data, goals, and baseline ASR method. Section 4 describes the results, which is followed by the conclusions in Section 5.

2. PROPOSED ENHANCEMENT APPROACH

The proposed enhancement approach takes M microphone signals as an input, finds the source signal containing speech among several environmental signals such as street and cafeteria noises, and enhances it by applying post-filtering on top of a beamformer output.

The ith microphone signal $x_i(k), i = 1, \ldots, M$ is modeled as a mixture of reverberated source signals $s_a(k), a = 1, \ldots, N_s$ embedded in additive noise $v_i(k)$ at time k. By taking the discrete time short-time Fourier transform (STFT), the model for the spectrum of the ith signal is written as

$$\mathbf{x}_i^{(t,n)} = \sum\nolimits_a \mathbf{s}_a^{(t,n)} \cdot \mathbf{h}_{i,a}^{(t,n)} + \mathbf{v}_i^{(t,n)}, \quad (1)$$

where $\mathbf{h}_{i,a}^{(t,n)}$ is an impulse response between the ith microphone signal $\mathbf{x}_i^{(t,n)}$ and source signal $\mathbf{s}_a^{(t,n)}$, $\mathbf{v}_i^{(t,n)}$ is noise signal, n is frequency index $n \in [0, N-1]$, and t is time frame index $t \in [0, T-1]$, where T is the amount of frames.

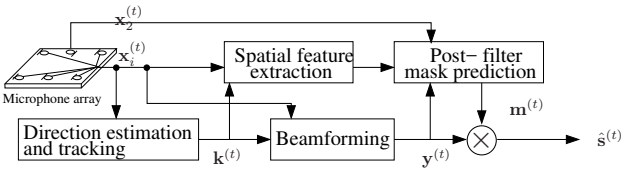


Fig. 1: A block diagram of the front-end enhancer is depicted. Speech source track $\mathbf{k}^{(t)}$ is used to steer a beamformer to obtain signal $\mathbf{y}^{(t)}$ at time t. Both spatial and spectral features are fed into the multilayer perceptron (MLP) to obtain a post-filter mask $\mathbf{m}^{(t)}$ that is then applied to beamformer output.

The block diagram of the proposed system is illustrated in Figure 1. The utilized 3rd CHiME Challenge data capturing device is a tablet where one of the six bezel mounted microphones is inverted to capture noise. Direction estimation and tracking is used to steer the delay-and-sum beamformer (DSB) and to extract spatial features. The beamformer output, spatial features, and the noise capturing microphone's signal are used to predict the post-filter mask. The predicted post-filter is then multiplied with the DSB output to produce the enhanced signal.

2.1. DoA estimation

The direction of arrival (DoA) estimation can be used to obtain a source direction at each time step to e.g. steer a beamformer. In the context of mobile multi-microphone device ASR, we are interested in the dominant speaker's DoA in the presence of background speech and non-speech sources.

The DoA vector is here defined in Cartesian coordinates $\mathbf{k} \in \mathbb{R}^3$ with respect to array origin. The relation to spherical coordinates (azimuth θ and elevation ϕ) is

$$\mathbf{k} = c^{-1} \left[\sin(\theta)\cos(\phi), \cos(\theta)\cos(\phi), \sin(\phi)\right]^{\mathrm{T}}, \quad (2)$$

where c is the speed of sound. The time difference of arrival (TDoA) between a microphone pair $\{i, j\}$ for a DoA vector $\mathbf{k}$ is defined as

$$\tau_{i,j}(\mathbf{k}) = (\mathbf{p}_i - \mathbf{p}_j)^{\mathrm{T}}\mathbf{k}, \quad (3)$$

where $\mathbf{p}_i \in \mathbb{R}^3$ denotes ith microphone location.

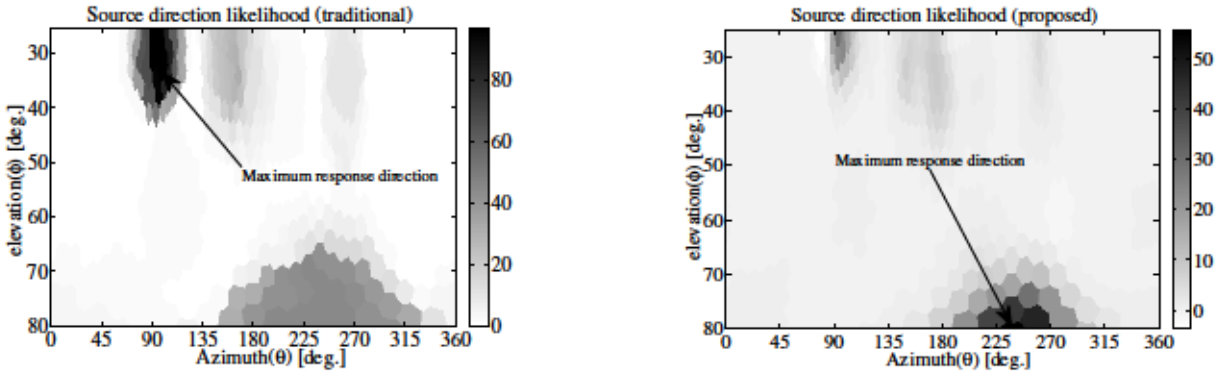
The generalized cross-correlation (GCC) is utilized in the source DoA estimation with the steered response power (SRP) method [33]

$$R(t, \mathbf{k}) = \sum_{\forall\{i,j\}} \sum_{n=0}^{N-1} \psi_{i,j}^{(t,n)} \cdot \mathbf{x}_i^{(t,n)} \cdot \overline{\mathbf{x}_j^{(t,n)}} \exp\left(j2\pi \frac{n}{N} \tau_{i,j}(\mathbf{k})\right), \quad (4)$$

where the outer summation is over all unique pairs $\{i, j\}$, $\overline{z}$ is complex conjugate of z, and $\psi_{i,j}^{(t,n)} = |\mathbf{x}_i^{(t,n)} \cdot \mathbf{x}_j^{(t,n)}|^{-1}$ is PHAT weighting that removes the amplitude information.

For simplicity, the speaker is assumed semi-static. Therefore, a grid of fixed DoA vectors $\mathbf{K} = [\mathbf{k}_1, \ldots, \mathbf{k}_{N_D}]^{\mathrm{T}}$ can be used to sum weighted SRP values over T frames

$$\Upsilon(\mathbf{k}) = \sum\nolimits_{t=0}^{T-1} \beta(t) \cdot R(t, \mathbf{k}), \quad (5)$$



(a) Traditional SRP-PHAT's marginal likelihood

(b) Marginal likelihood of speech weighted SRP-PHAT

Fig. 2: The outputs of traditional SRP-PHAT (panel (a)) and proposed speech weighted SRP-PHAT (panel (b)) are depicted for a speech signal captured on a bus. Actual speaker is located in direction ($\theta = 250°, \phi = 80°$), which is the global maximum of the proposed method but only a local minimum of the traditional approach.

where $\beta(t)$ represents the frame weight. Traditional SRP utilized $\beta(t) = 1$, and frame log-energy has been used in [30]. The DoA over the sequence can be estimated by $\hat{\mathbf{k}} = \text{argmax}_{\mathbf{k}\in\mathbf{K}}\{\Upsilon(\mathbf{k})\}$. The speech weighted steered response power (SW-SRP) is proposed by setting

$$\beta(t) = \alpha_1 w_{\text{s}}(t) + \alpha_2 w_{\text{s+bg}}(t) + \alpha_3 w_{\text{bg}}(t), \qquad (6)$$

where $w_{\text{s}}(t), w_{\text{s+bg}}(t)$ and $w_{\text{bg}}(t)$ are framewise likelihood values for clean speech, speech in background, and background, and $\alpha_1, \ldots, \alpha_3$ are class weights here set to $2, 1, -1$, correspondingly. The likelihood values are outputs of a MLP to be described in section 2.2. Using the proposed weight values for $\alpha_1, ..., \alpha_3$ emphasizes directions containing speech in the SRP output and de-emphasizes the non-speech directions. In the presence of multiple speech signals, the direction with more clean speech is emphasized over directions with speech in background noise. Figure 2 illustrates the effect of weighting for the response from a sentence in noise. The left panel (a) is the output of traditional SRP-PHAT, where the maximum response direction ($\theta = 100°, \phi = 30°$) contains non-speech. The right panel (b) displays the output of SW-SRP-PHAT, where the maximum value corresponding to the actual speaker direction ($\theta = 250°, \phi = 80°$).

2.2. Speech vs. background likelihood estimation

An MLP is trained to classify microphone signal frames to three different classes (speech, speech in background, and background). The MLP uses MFCC and amplitude modulation spectrum (AMS)[1] and Δ-features. 30 MFCCs and 75 AMS features are used. The AMS feature extraction here uses 5 frequency bands, and each band consists of 15 modulation amplitudes [17]. The window length is 512 samples with 50 % overlap.

The input and hidden layer dimensions are 209 with the 1st MFCC component removed, and the output layer consists of three nodes. Each node's activation function is of sigmoid type that has a value range between [0,1], which is suitable for likelihood interpretation. The 3rd CHiME Challenge data, to be described in Sect. 3.1, is used for training. Lapel microphones are used to provide features for the speech class, tablet microphone #1 is used to obtain features for "speech in background" class, and pauses between sentences from the embedded data are used to obtain background features.

Figure 3 illustrates the operation of the speech likelihood estimation. The input signal (top panel) is used to extract features for the MLP, which has three output nodes for the classes: w_{s} for speech, $w_{\text{s+bg}}$ speech in background noise, and w_{bg} for background. The output activity of each node is visualized for the input signal in Fig. 3, where w_{s} and $w_{\text{s+bg}}$ are plotted in the second panel from the top. The third panel from top depicts the outputs of the background node w_{bg}. The bottom panel depicts the lapel signal for reference purposes.

2.3. Tracking

Tracking is performed to estimate speaker's state by considering the DoA measurements sequentially and by rejecting possible outliers.

With the traditional SRP-PHAT approach in a context with relatively loud continuous environmental noises, one would need to track multiple sources to include the speaker in the set of tracked sources. Since the global maximum of the SW-SRP-PHAT is expected to be related to the speaker direction, the need for multiple target tracking (MTT) can be avoided. This allows reduced complexity of tracking and elimination of the assignment problem of measurements to tracks. Furthermore, even after the successful tracking of multiple targets, one would still need to identify the speech source to be enhanced and transcribed. To accommodate for small source direction variation, a sequential Bayesian estimator is applied for tracking of the temporally evolving source direction. More specifically, the Kalman filter is used. The state and measurements equations can be written for the

[1]Matlab software for AMS extraction `http://ecs.utdallas.edu/loizou/AMS_Binary_Mask_Demos/demos.html`

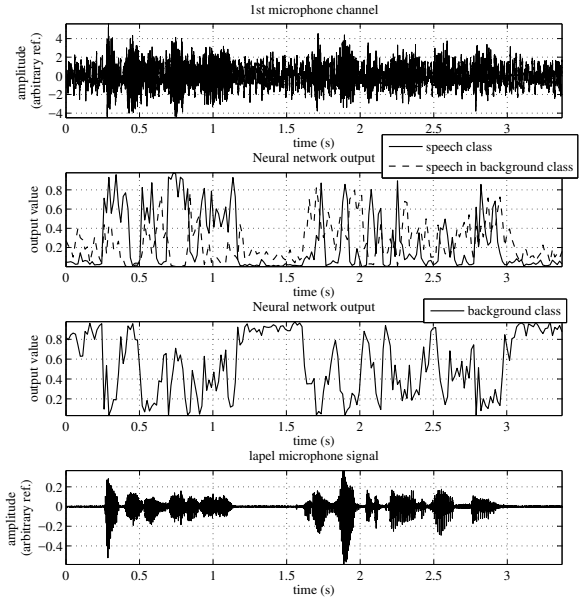


Fig. 3: The operation of the speech likelihood estimation.

DoA estimation problem as [34]

$$\mathbf{b}^{(t)} = \mathbf{A}\mathbf{b}^{(t-1)} + \mathbf{q}^{(t)}, \quad (7)$$

$$\mathbf{z}^{(t)} = \mathbf{H}\mathbf{b}^{(t)} + \mathbf{r}^{(t)}, \quad (8)$$

where $\mathbf{A}$ is the transition matrix of the dynamic model and $\mathbf{b}^{(t)}$ denotes the source state that consists of direction $\mathbf{k}$, corresponding velocity $\dot{\mathbf{k}}$, and acceleration values $\ddot{\mathbf{k}}$. $\mathbf{H}$ is the observation matrix that maps the state estimate into a measurement vector, and $\mathbf{q}^{(t)} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ and $\mathbf{r}^{(t)} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ are Gaussian noise processes at time t with covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, respectively. The measurement at each time step consists of the instantaneous DoA estimate $\hat{\mathbf{k}}^{(t)} = \text{argmax}_{\mathbf{k}\in\mathbf{K}}\{R(t,\mathbf{k})\}$ while the track is initialized into the direction of maximum SW-SRP-PHAT $\Upsilon(\mathbf{k})$, in Eqs. (5) and (6). Since the instantaneous measurements contain directions also from other sources, source direction measurements $\hat{\mathbf{k}}^{(t)}$ with larger angle deviation than $(20°)$ from initial direction are discarded as outliers. The output of the tracking process is the smoothed speaker trajectory $\mathbf{k}^{(t)}$ at each time step t.

2.4. Beamforming with post-filtering

In order to amplify the speaker direction and reduce noise, the DSB temporally aligns the microphone signals before summation. The DSB output is multiplied with a spatial post-filter to further attenuate unwanted source directions

$$\begin{aligned}\hat{\mathbf{s}}^{(t,n)} &= \frac{1}{M}\sum\nolimits_{i=1}^{M} \mathbf{x}_i^{(t,n)} \exp\left(j2\pi\frac{n}{N}\tau_{i,1}(\mathbf{k}^{(t)})\right) \cdot \mathbf{m}^{(t,n)} \\ &= \mathbf{y}^{(t,n)} \cdot \mathbf{m}^{(t,n)} \end{aligned} \quad (9)$$

where $\mathbf{y}^{(t,n)}$ represents the DSB output for the speaker at time t in direction $\mathbf{k}^{(t)}$, and $\mathbf{m}^{(t,n)}$ is the corresponding post-filter.

2.5. Post-filter training and prediction

A neural network is used to learn how phase and amplitude based features with temporal information extracted from the microphones can be used to predict a post-filter.

Given a source direction $\mathbf{k}^{(t)}$ (refer to Section 2.3), the target's spatial feature $\mathbf{u}^{(t,n)}$ is obtained by averaging the cosine of angle differences between the measured and theoretical phase differences over all P microphone pairs $\{i,j\}$ [30]

$$\mathbf{u}^{(t,n)} = \frac{1}{P}\sum_{\forall\{i,j\}} \cos\Big(\angle\mathbf{x}_i^{(t,n)} - \angle\mathbf{x}_j^{(t,n)} - 2\pi\frac{n}{N}\tau_{i,j}(\mathbf{k}^{(t)})\Big), \quad (10)$$

where $\angle z$ denotes phase of complex value z. To include spatial information about background, cues from directions equally spaced on a sphere, omitting angles close to the source, are calculated using (10) and then averaged. The spatial feature vectors of the target and background are decimated to 30 mel bands and concatenated. In addition, 30 MFCCs from the DSB output $\mathbf{y}^{(t)}$ and from the noise capturing microphone $\mathbf{x}_2^{(t)}$ are included as features to add more information about the target and background spectra. The concatenated feature vector's dimension is 120. Finally, to include temporal information, the feature vectors from frames $t-1$, and $t+1$ are concatenated to the tth frame's feature vector. The final input feature vector dimension is 360.

The target vector of the neural network is based on the Wiener filter [35]

$$\mathbf{m}_{\text{WF}}^{(t,n)} = \frac{\mathbf{p}_S^{(t,n)}}{\mathbf{p}_S^{(t,n)} + \mathbf{p}_N^{(t,n)}}, \quad (11)$$

where $\mathbf{p}_S^{(t,n)}$ and $\mathbf{p}_N^{(t,n)}$ denote the signal and noise power spectra, respectively. The noise power spectra here also include the unwanted sources. The target values are also decimated into 30 mel bands. WF values near zero represent the majority of values. We note that mask prediction errors close to zero affect the attenuation of unwanted sounds more than mask prediction errors near value one. Therefore, to further emphasize the attenuating capability of the post-filtering, the WF values are transformed non-linearly with traditional μ-law compression with $\mu = 100$ to obtain a finer scale for values near zero. In the enhancement stage, the output values first undergo an inverse μ-law compression and are then transformed to linear frequency scale using an inverse mel-scale transform before being applied to DSB output (9).

Post-filter prediction training was conducted using the 3rd CHiME Challenge data, discussed later in Section 3.1 and detailed in [32]. The training data consisted of the provided real and simulated recordings of the training data. Target signal power spectrum $\mathbf{p}_S^{(t,n)}$ is obtained from the lapel microphone in the case of real recordings, or from the original signal in the case of simulations. The noisy input signal spectrum $\mathbf{p}_N^{(t,n)}$ is obtained from the tablet's noise capturing microphone. Training data contains 1600 real and 7138 simulated utterances.

Table 1: Objective scores for unenhanced data (UN), beamformed data (DSB), and DSB with proposed post-filtering (PF).

Dataset	Training						Development						Evaluation					
Type / files	Real / 1600			Simulated / 7138			Real /1640			Simulated /1640			Real/1320			Simulated/1320		
	UN	DSB	PF	UN	DSB	PF	UN	DSB	PF	UN	DSB	PF	UN	DSB	PF	UN	DSB	PF
STOI	0.55	0.59	0.65	0.82	0.89	0.92	0.51	0.56	0.61	0.64	0.69	0.71	0.41	0.47	0.53	0.56	0.68	0.70
fwSNRseg	0.1	4.1	6.7	1.4	3.5	10.7	-1.4	2.5	6.4	-1.0	0.7	5.7	1.6	5.6	5.5	-3.0	0.0	4.8
SNRseg	-2.7	1.8	4.0	-0.1	2.8	6.9	-4.7	2.5	3.6	-3.8	-1.0	2.9	-1.1	3.2	3.3	-5.8	-2.0	2.0
SRMR	2.7	3.6	5.2	3.8	4.8	6.0	2.1	3.0	4.5	3.3	3.9	5.1	2.1	3.0	5.1	2.4	5.2	7.0

Training was done using Python 2.7 with scikit-neuralnetwork v.0.2 [36]. The feature processing included scaling of the input features using standard deviation. The hidden layer of the MLP is of the same dimension and type as the input layer (360 nodes, with $\tanh()$ activation function). The output layer consists of 30 sigmoid nodes corresponding to the μ-law compressed mel band mask values. L2-norm of the layers weights multiplied by 10^{-4} was added to regularize MLP to reduce overfitting. 95% of post-filter training data was used for training of prediction, and 5% for testing of prediction error. Stochastic gradient descent was used with mini-batch size of 50 samples with learning rate of 0.02. The training was stopped after 5 epocs since the error did not decrease (MSE 0.016).

3. EVALUATION ON 3RD CHIME CHALLENGE DATA

3.1. Data

A brief overview of 3rd CHiME Challenge data is presented. For a complete description, refer to [32]. The data consists of spoken utterances from a 5000-word vocabulary subset of Wall Street Journal (WSJ) corpus and is captured using a tablet with 6 microphones, sampled at 16kHz and at 16-bit depth. The data is is divided into training (TR), development (DT), and evaluation (ET) sets, each set containing both real and simulated recordings. Real recordings consist of speech recorded in 4 different noisy environments viz., public transport (BUS), street (PED), cafeteria (CAF), and street (STR).

3.2. Preprocessing

Each sentence is processed as a separate recording. Occasionally, due to microphone failures, a recording contains missing values of data in one or more channels. Channels with large portions of missing values are detected using an energy based threshold and omitted.

During the post-filter training, the lapel microphone is aligned to the first microphone channel to remove an occasionally occurring time offset between channels [32]. This is done to avoid a mismatch between the features obtained from the tablet microphones and the post-filter target value obtained using also the lapel microphone.

3.3. Enhancement

The proposed enhancement method described in Section 2 is applied to the noisy microphone signals using the set of isolated recordings to produce an enhanced signal for the recognizer. Processing window length is 512 samples with 50 % overlap. The number of fixed DoAs in Sect. 2.1 is $N_D = 731$. The DoAs are equally spread on the tablet's upper hemisphere, i.e., the side with five of the six microphones facing forward. In order to omit angles that are impractical for reading a tablet's screen, angles more than 65° apart from to tablet surface's normal are removed.

3.4. Recognizer

The ASR baseline is provided by the organizers and consists of GMM and DNN based recognizers from Kaldi toolkit. A general outline is given in this section. For more specific details of the recognizers, refer to [32]. In these experiments, the proposed system only acts as a front-end with matched enhancement used for acoustic model training and testing.

The GMM baseline comprises triphone based acoustic models with MFCCs as acoustic features. It also includes various feature transformations such as Linear Discriminant Analysis (LDA), Maximum likelihood linear transformation (MLLT) and feature space maximum likelihood linear regression (MLLR) with speaker-adaptive training.

Out of the two DNN variants available in Kaldi toolkit, Karel's setup is provided by the organizers. Its performance is reported as state-of-the-art [32]. The network is pre-trained by restricted Boltzmann machines, followed by cross-entropy training, and sequence discriminative training.

4. RESULTS

4.1. Objective scores for signal enhancement

Four objective scores are calculated and reported for the enhanced audio. Segmental SNR (SNRseg) and the frequency-weighted SNR (fwSNRseg[2]) are spectral distance metrics [37]. The output was additionally analyzed with the short-time objective intelligibility measure (STOI) [38], which is designed for predicting the intelligibility of TF weighted noisy speech. Finally, non-invasive speech-to-reverberation modulation energy ratio (SRMR) metric is reported. It evaluates speech quality and intelligibility based on a modulation spectral representation of the speech signal [39].

[2] fwSNRseg uses 25 mel bands, weight parameter $\gamma = 0.2$.

Table 2: Average WERs (%) for 3rd CHiME Challenge data sets using unenhanced data, baseline enhancement, and the proposed system with GMM and DNN back-ends. Training and test data are always matched.

Model	Data	Development		Evaluation	
		Real	Sim.	Real	Sim.
GMM	unenhanced	18.70	18.71	33.23	21.59
	baseline MVDR	20.55	9.79	37.36	10.59
	proposed DSB+PF	11.69	14.78	22.39	21.54
DNN	unenhanced	16.13	14.30	33.43	21.51
	baseline MVDR	17.72	8.17	33.76	11.19
	proposed DSB+PF	10.52	13.58	21.10	24.39

Table 1 presents the results for the proposed enhancement method averaged over the different environments. The unenhanced signal and DSB output are also scored for reference. The unenhanced signal has the lowest scores in all cases. The post-filter provides improvement over DSB for STOI and for SNR scores in all cases except for the real evaluation data, where the SNRs are equal. In addition, the SRMR is consistently higher with post-filtering. The results agree with [30].

4.2. Performance of speech/background classifier

The implemented speech classification into three classes (speech, speech in background, background) is evaluated based on balanced F1 score. The micro-averaged and macro-averaged F1 scores are observed both to be 0.95. Without the AMS features (using only the MFCC + ΔMFCC) the corresponding F1 scores are 0.92.

4.3. Speech recognition accuracy

The results for ASR are reported as word error rates (WERs) in Tables 2 and 3. Table 2 lists average WERs over all environments for each data sets using original unenhanced data, provided baseline enhancement, and the proposed system. The provided baseline is MVDR beamforming with sample covariance method utilizing 400 to 800 ms of background before and after each sentence with diagonal loading [32]. The steering uses Viterbi decoding for a grid of SRP-PHAT values, and assumes that speech sources are most probably at 90° elevation. It is pointed out that we utilize the isolated sentences which generally do not contain as much background before and after each sentence.

All results are evaluated using GMM and DNN back-ends. Baseline scores are replicated from [32], although effectively equal results were also achieved in our own evaluations. Table 3 lists detailed scores of the chosen system for each set and environment using the DNN back-end.

4.4. Discussion

According to the results, the system manages to improve the objective quality of all sets. However, the ASR results show more inconsistent behavior between original data, baseline enhancement, and the proposed method.

Table 3: WERs (%) of the proposed system for each environment and test set using the Kaldi DNN back-end.

Environment	Development		Evaluation	
	Real	Sim.	Real	Sim.
BUS	11.86	11.43	28.38	16.31
CAF	10.35	15.71	20.60	22.81
PED	8.4	11.00	19.00	26.90
STR	11.46	16.17	16.42	31.57

The challenge baseline enhancement yields major WER improvements for simulated data, but completely fails to improve the performance for authentic real-world recordings even by using the state-of-the-art DNN back-end. This suggests that the system may be able to invert some of the simulation process, which appears very promising for such data, yet does not generalize to actual multichannel recordings.

Conversely, the proposed system shows even detrimental performance for simulated data, but yields uniform and significant improvements for real recordings, reducing the WER there by 34–40% (relative) compared to unenhanced data and the largely ineffective baseline enhancement. Because the major goal of robust ASR research is to improve its performance in actual use cases, the results can be considered positive. Nevertheless, it should be investigated, whether the simulated mixing process is truly useful, because the evaluated systems show heavy and conflicting biases toward opposite sets. As a majority of audio material used for back-end training is simulated, the bias will also affect the main recognizer performance. Note that the DNN back-end was not re-trained after the final small system changes, thus the GMM train-test match can be considered slightly more reliable.

5. CONCLUSIONS

A multichannel system was proposed to separate speech from real-world noises in order to improve the accuracy of automatic speech recognition. The dominant speaker's direction is extracted with proposed speech weighted steered response power (SW-SRP). The direction is used to initialize a tracker that steers the beamformer. A predicted post-filter is then applied to the beamformed signal to further reduce noise. Results were evaluated using the 3rd CHiME Challenge corpus, where medium vocabulary utterances are spoken in everyday environments and recorded using a tablet device with six microphones. The proposed system managed to reduce the word error rate of real recordings by up to 40%. However, conflicting results were observed for simulated data, calling for further investigation on the simulation process and its effect on the overall framework. Despite these issues, the results suggest that multichannel enhancement can act as a crucial component in robust real-world speech processing systems.

6. REFERENCES

[1] T. Virtanen, R. Singh, and B. Raj, Eds., *Techniques for Noise Robustness in Automatic Speech Recognition*, Wiley, New York, NY, USA, 2012.

[2] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The Second CHiME Speech Separation and Recognition Challenge: an Overview of Challenge Systems and Outcomes," in *Proc. ASRU*, Olomouc, Czech Republic, 2013, pp. 162–167.

[3] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An Overview of Noise-Robust Automatic Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[4] R. M. Stern and N. Morgan, "Features Based on Auditory Physiology and Perception," in *Techniques for Noise Robustness in Automatic Speech Recognition*, T. Virtanen, R. Singh, and B. Raj, Eds. Wiley, New York, NY, USA, 2013.

[5] A. Hurmalainen, J. F. Gemmeke, and T. Virtanen, "Compact Long Context Spectral Factorisation Models for Noise Robust Recognition of Medium Vocabulary Speech," in *Proc. 2nd International Workshop on Machine Listening in Multisource Environments (CHiME)*, Vancouver, Canada, 2013, pp. 13–18.

[6] E. Yılmaz, J. F. Gemmeke, and H. Van hamme, "Noise Robust Exemplar Matching Using Sparse Representations of Speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 8, pp. 1306–1319, 2014.

[7] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[8] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng, "Recurrent Neural Networks for Noise Reduction in Robust ASR," in *Proc. 13th Annual Conference of the International Speech Communication Association (Interspeech)*, Portland, OR, USA, 2012, pp. 22–25.

[9] A. Narayanan and D. Wang, "Ideal Ratio Mask Estimation Using Deep Neural Networks for Robust Speech Recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, 2013, pp. 7092–7096.

[10] F. Weninger, F. Eyben, and B. Schuller, "Single-Channel Speech Separation With Memory-Enhanced Recurrent Neural Networks," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 3709–3713.

[11] J. T. Geiger, F. Weninger, A. Hurmalainen, J. F. Gemmeke, M. Wöllmer, B. Schuller, G. Rigoll, and T. Virtanen, "The TUM+TUT+KUL Approach to the CHiME Challenge 2013: Multi-Stream ASR Exploiting BLSTM Networks and Sparse NMF," in *Proc. 2nd International Workshop on Machine Listening in Multisource Environments (CHiME)*, Vancouver, Canada, 2013, pp. 25–30.

[12] J. T. Geiger, F. Weninger, J. F. Gemmeke, M. Wöllmer, B. Schuller, and G. Rigoll, "Memory-enhanced neural networks and NMF for robust ASR," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 6, pp. 1037–1046, 2014.

[13] D. Wang, "On Ideal Binary Mask As the Computational Goal of Auditory Scene Analysis," in *Speech Separation by Humans and Machines*, P. Divenyi, Ed., chapter 12. Kluwer Academic Publishers, 2005.

[14] S. Srinivasan, N. Roman, and D. Wang, "Binary and Ratio Time-Frequency Masks for Robust Speech Recognition," *Speech Communication*, vol. 48, no. 11, pp. 1486–1501, 2006.

[15] C. Hummersone, T. Stokes, and T. Brookes, "On the Ideal Ratio Mask as the Goal of Computational Auditory Scene Analysis," in *Blind Source Separation: Advances in Theory, Algorithms and Applications*, Ganesh R Naik and Wenwu Wang, Eds., chapter 12. Springer, 2014.

[16] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2067–2080, Sept 2011.

[17] G. Kim, Y. Lu, Y. Hu, and P. C. Loizou, "An Algorithm that Improves Speech Intelligibility in Noise for Normal-Hearing Listeners," *The Journal of the Acoustical Society of America*, vol. 126, no. 3, pp. 1486–1494, 2009.

[18] Y. Wang and D. Wang, "Towards Scaling Up Classification-Based Speech Separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1381–1390, July 2013.

[19] Ö. Yılmaz and S. Rickard, "Blind Separation of Speech Mixtures via Time-Frequency Masking," *IEEE Transactions on Signal Processing*, vol. 7, no. 52, pp. 1830–1847, July 2004.

[20] N. Roman, D. Wang, and G. J. Brown, "Speech Segregation based on Sound Localization," *J. Acoust. Soc. Am.*, vol. 114, no. 4, pp. 2236–2252, 2003.

[21] D. Ayllon, R. Gil-Pita, and M. Rosa-Zurera, "Rate-Constrained Source Separation for Speech Enhancement in Wireless-Communicated Binaural Hearing Aids," *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, 2013.

[22] J. Woodruff and D. Wang, "Binaural Detection, Localization, and Segregation in Reverberant Environments based on Joint Pitch and Azimuth Cues," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 806–815, 2013.

[23] Y. Jiang, D. Wang, and R. Liu, "Binaural Deep Neural Network Classification for Reverberant Speech Segregation," in *Proc. 15th Annual Conference of the International Speech Communication Association (Interspeech)*, 2014.

[24] I. J. Tashev, *Sound Capture and Processing: Practical Approaches*, John Wiley & Sons Ltd., 2009.

[25] C. Zhang, D. Florencio, and Z. Zhang, "Why does PHAT work well in lownoise, reverberative environments?," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 2565–2568.

[26] E. A. Lehmann and A. M. Johansson, "Particle filter with integrated voice activity detection for acoustic source tracking," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 28–28, 2007.

[27] M. Kepesi, F. Pernkopf, and M. Wohlmayr, "Joint position-pitch tracking for 2-channel audio," in *Proc. International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2007, pp. 303–306.

[28] G. W. Pulford, "Taxonomy of multiple target tracking methods," *IEE Proceedings Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 291–304, October 2005.

[29] P. Swietojanski, A. Ghoshal, and S. Renals, "Hybrid Acoustic Models for Distant and Multichannel Large Vocabulary Speech Recognition," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013, pp. 285–290.

[30] P. Pertilä and J. Nikunen, "Microphone Array Post-Filtering Using Supervised Machine Learning for Speech Enhancement," in *Proc. 15th Annual Conference of the International Speech Communication Association (Interspeech)*, 2014.

[31] P. Pertilä and J. Nikunen, "Distant speech separation using predicted timefrequency masks from spatial features," *Speech Communication*, vol. 68, pp. 97 – 106, 2015.

[32] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' Speech Separation and Recognition Challenge: Dataset, task and baselines," in *Submitted to IEEE 2015 Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015.

[33] J. H. DiBiase, H. F. Silverman, and M. S. Brandstein, "Robust Localization in Reverberant Rooms," in *Microphone Arrays*, M. Brandstein and D. Ward, Eds., chapter 8, pp. 157–180. Springer-Verlag, 2001.

[34] S. Särkkä, *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013.

[35] E. Diethorn, "Subband Noise Reduction Methods for Speech Enhancement," in *Audio Signal Processing for Next-Generation Multimedia Communication Systems*, Y.(A.) Huang and J. Benesty, Eds., chapter 4, pp. 91–115. Kluwer Academic Publishers, 2004.

[36] A. Champandard, "scikit-neuralnetwork," June 2015, https://github.com/aigamedev/scikit-neuralnetwork.

[37] Y. Hu and P. C. Loizou, "Evaluation of Objective Quality Measures for Speech Enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 229–238, Jan 2008.

[38] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An Algorithm for Intelligibility Prediction of Time–Frequency Weighted Noisy Speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011, (Software: `http://www.ceestaal.nl/stoi.zip`).

[39] T. H. Falk, C. Zheng, and W.-Y. Chan, "A non-intrusive quality and intelligibility measure of reverberant and dereverberated speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1766–1774, Sept 2010, (Software: `http://github.com/MuSAELab/SRMRToolbox`).